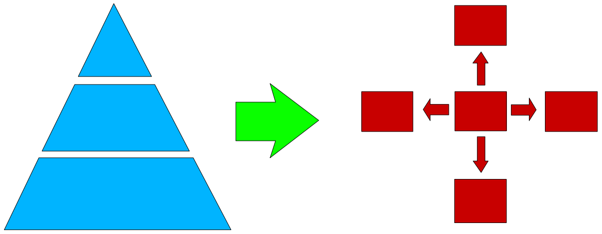
Sessions 3215 and 3216
Practical Guide to Optimizing APPN and EBN Searches
(Parts 1 and 2)



Johnathan Harter
z/OS CommServer Development
yoda@us.ibm.com

SHARE 2009 Summer
Technical Conference

Agenda

- Searching using ALIAS names
- Who is my network node server (NNS)?
- Introduction to Subarea and APPN searching
- Tuning Subarea Searching
  - Subarea search controls
  - ADJSSCP tables for sample subarea-to-APPN migration
- Tuning APPN EBN Searching
  - Introduction to extended border nodes (EBNs)
  - APPN search controls
  - Subnetwork routing lists (SRLs)
  - ADJCLUST tables for sample mixed APPN and subarea network
- Hints to aid in debugging search problems

Why avoid searching using ALIAS names?

- ALIAS names used for resources whose NETID is unknown
  - "Default" ADJSSCP and/or ADJCLUST tables are used for searching
  - Typical "default" ADJSSCP/ADJCLUST tables allow maximum search scope
    - That is, they are defined to search everywhere
  - EBN searches for ALIAS names treat BNDYN=LIMITED like BNDYN=FULL

- Performance problems can result from ALIAS searching
  - Additional CPU required to process unnecessary searches or search steps
  - Network bandwidth required to send/receive search requests and responses
  - Does not matter whether the search originated natively or non-natively

- Performance problems for non-native networks too!
  - Adjacent (SNI and EBN) networks can be affected by ALIAS searches too!
  - Searches originating in adjacent (SNI and EBN) networks can affect native network!

How do I prevent ALIAS name searches?

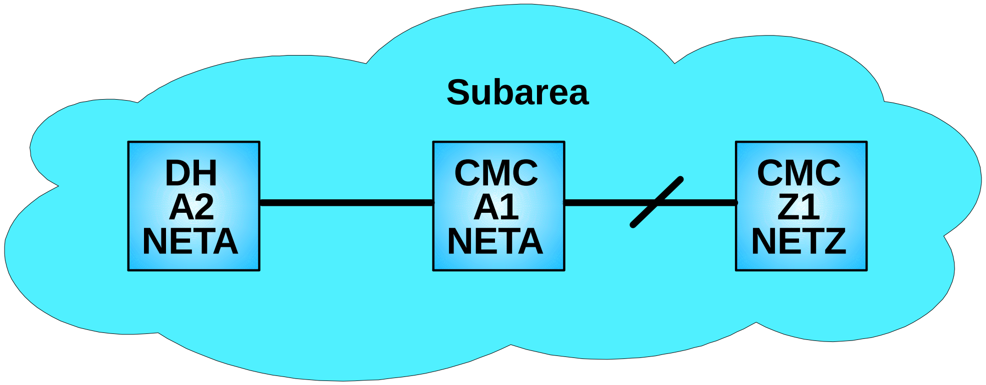
- Convert ALIAS names to network-qualified real names
  1. Use ALIAS translation function of session management exit (SME)
  2. Use NetView ALIAS translation application
  3. Predefine network-qualifed (real) CDRSCs
     - Code NETID=netid on preceding NETWORK statement
     - Only works if the ALIAS name is the same as the (unqualified) real name

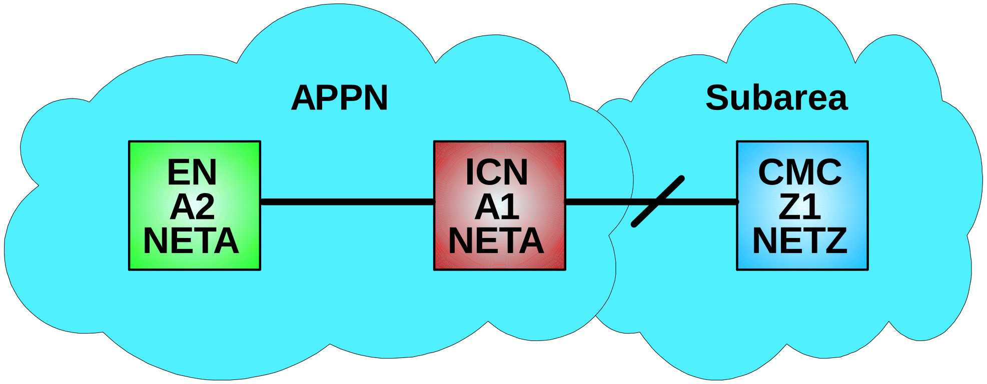
- z/OS CS V1R8: ALIASRCH=YES or NO on ADJCPs
  1. Only allowed for ADJCPs that define non-native NNs or EBNs
  2. Controls whether ALIAS searches are allowed from the ADJCP being defined
  3. Provides granular control over ALIAS searches entering your network

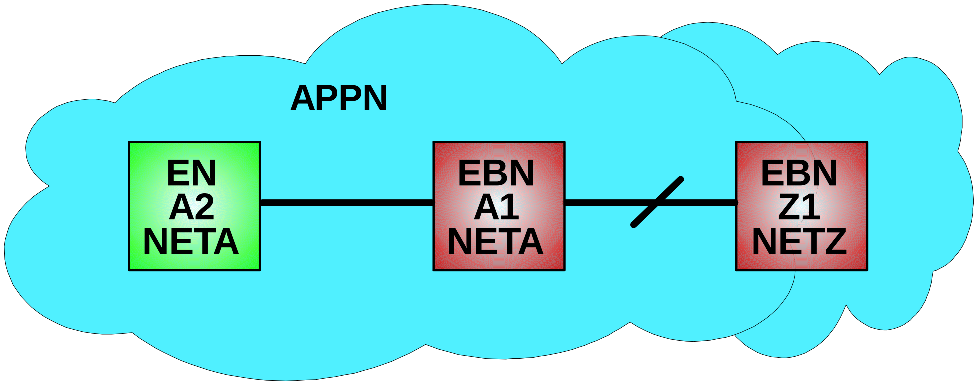
- z/OS CS V1R10: AUTHNETS= on ADJCPs
  1. Only allowed for ADJCPs that define non-native NNs or EBNs
  2. Controls network-qualified searches received from the ADJCP being defined
     - Default (AUTHNETS not coded) allows network-qualified searches for all NETIDs
     - AUTHNETS=NETA allows only netid-qualified searches for target LUs in NETA
     - AUTHNETS=(NETA,NETB,NETC) allows only netid-qualified searches for target LUs in NETA, NETB or NETC
     - AUTHNETS= rejects all network-qualified searches received from this ADJCP

ALIAS name translation: subarea versus APPN

- Pure subarea and SNI
  - ALIAS translation can be done at the network boundary (CMC/GWSSCP)
    - Search traverses subarea side
- Network boundary is SNI
  - ALIAS translation can still be done at network boundary (CMC/ICN)
    - Search traverses subarea side
  - But non-optimal EBN searching may occur first (on APPN side)!
- Pure APPN and EBN
  - ALIAS translation **CANNOT** be done at network boundary (CMC/EBN)
    - Search is not seen by subarea side
  - To optimize searching, ALIAS translation must occur on EN instead





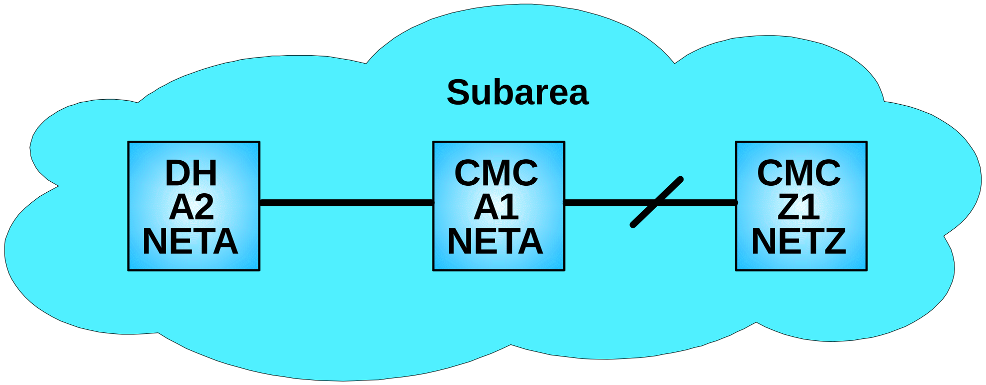


Strategies for predefining CDRSCs

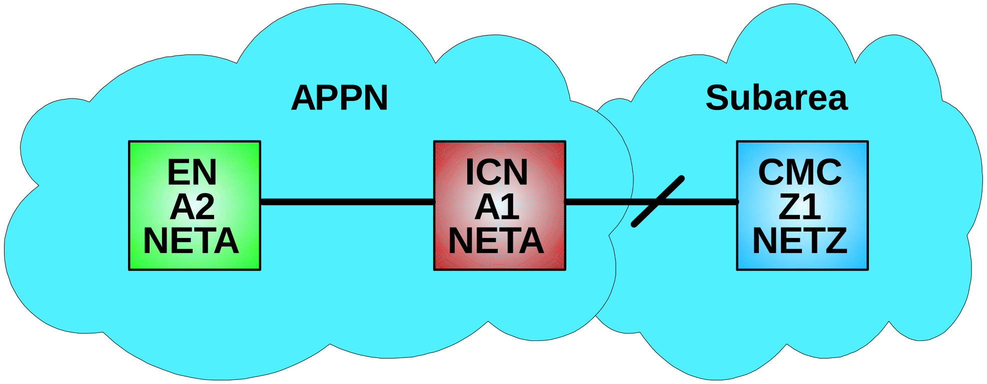
- Predefine ALL resources as CDRSCs with NETID
  - "Default" ADJSSCP/ADJCLUST tables are no longer needed
    - All searches should contain the network-qualifed (real) target LU name
  - Can be an administrative nightmare (every new resource must be predefined)
- Predefine ALL cross-network resources
  - "Default" ADJSSCP/ADJCLUST tables should search only the native network(s)
    - All searches containing ALIAS names should find only native resources
  - Define appropriate ADJSSCP/ADJCLUST tables for non-native NETIDs
    - All searches for non-native resources should contain the network-qualifed (real) target LU name
- Where should predefined CDRSCs be activated?
  - Subarea: Probably only required on CMCs and/or GW SSCPs
  - APPN: May be required on ALL hosts (NNs, ICNs, ENs and MDHs)
    - With "NQNMODE For DS DB Entries" in z/OS CS V1R4:
      - Predefined CDRSCs may only be required on NNs and ICNs
      - Requires NQNMODE=NAME (specified or defaulted) and CPNAME=

Predefined CDRSCs: Subarea Versus APPN

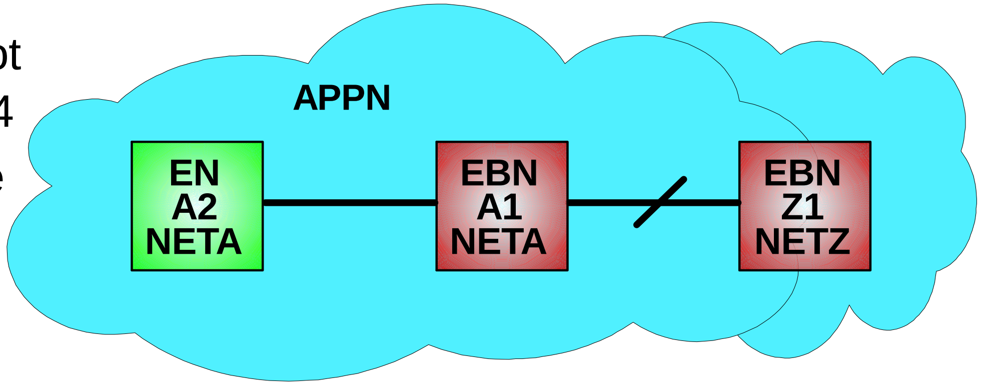
- Pure subarea and SNI
  - CDRSCs defined on CMC/GWSSCP can be used for sessions from DH
    - Search traverses subarea side



- Network boundary is SNI
  - CDRSCs defined on CMC/ICN can be used for sessions from DH
    - Search traverses subarea side
  - But non-optimal EBN searching may occur first (on APPN side)!



- Pure APPN and EBN
  - CDRSCs defined on CMC/EBN are not used for sessions from ENs until V1R4
    - Search is not seen by subarea side
  - V1R4: CDRSCs defined on NNs with NQNMODE=NAME and CPNAME= can be used for EN searches too!



ALIAS names recommendations

- Convert ALIAS names to network-qualified real names as close to the OLU host as possible!
  - When possible, predefine CDRSCs (ALIAS name must be same as real name)
    - For subarea networks: on CMC or GW SSCP
    - For APPN networks: on OLU host or OLU NNS
  - If necessary, implement SME or NetView ALIAS translation
    - For subarea networks: on CMC or GW SSCP
    - For APPN networks: on OLU host (which probably means EVERY host)
  - If you already have a DSME, consider enhancing it to convert non-authentic NETIDs to authentic NETIDs
    - V1R7 (with APAR OA12411) or later
- Optimizes searching and minimizes overhead by allowing all search steps to be performed based on the real NETID of the target resource

Agenda

- Searching using ALIAS names
- Who is my network node server (NNS)?
- Introduction to Subarea and APPN searching
- Tuning Subarea Searching
  - Subarea search controls
  - ADJSSCP tables for sample subarea-to-APPN migration
- Tuning APPN EBN Searching
  - Introduction to extended border nodes (EBNs)
  - APPN search controls
  - Subnetwork routing lists (SRLs)
  - ADJCLUST tables for sample mixed APPN and subarea network
- Hints to aid in debugging search problems

Understand who your network node server is!

- Why is it so important to understand who my network node server (NNS) is?
  - Optimizing search performance means avoiding broadcast searches
    - Directed searches consume less CPU and network bandwidth
  - Directed searches are always sent to the target LU's NNS
    - NNs are only aware of adjacent ENs, so ...
    - Not all ENs are known to all NNs (local versus network topology)
  - Predefining CDRSCs with NETSRVR and/or CPNAME can help
    - Especially if "NQNMODE For DS DB Entries" is used

Who is my network node server (NNS)?

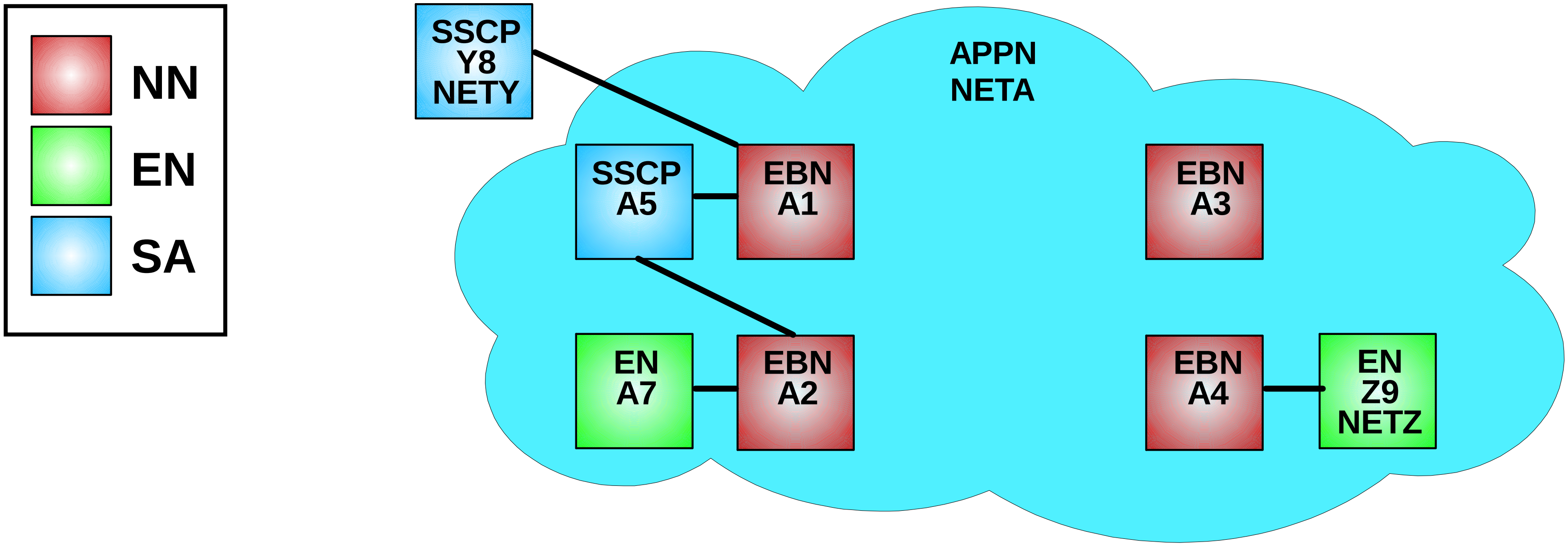
- Network nodes are their own NNS

- End nodes use native NN as their NNS
  - EN's NETID does not have to match NETID of NNS
  - ENs can change NNS at their discretion
    - After link INOP or CP-CP session failure
    - Due to NNSPREF start option

- Subarea nodes use attaching ICN as their NNS
  - Subarea nodes are treated as ENs served by Interchange Nodes (ICNs)
    - Subarea node's NETID does not have to match NETID of ICN/NNS
    - Works for same-network or cross-network subarea nodes
  - Subarea nodes can appear to have multiple NNSs

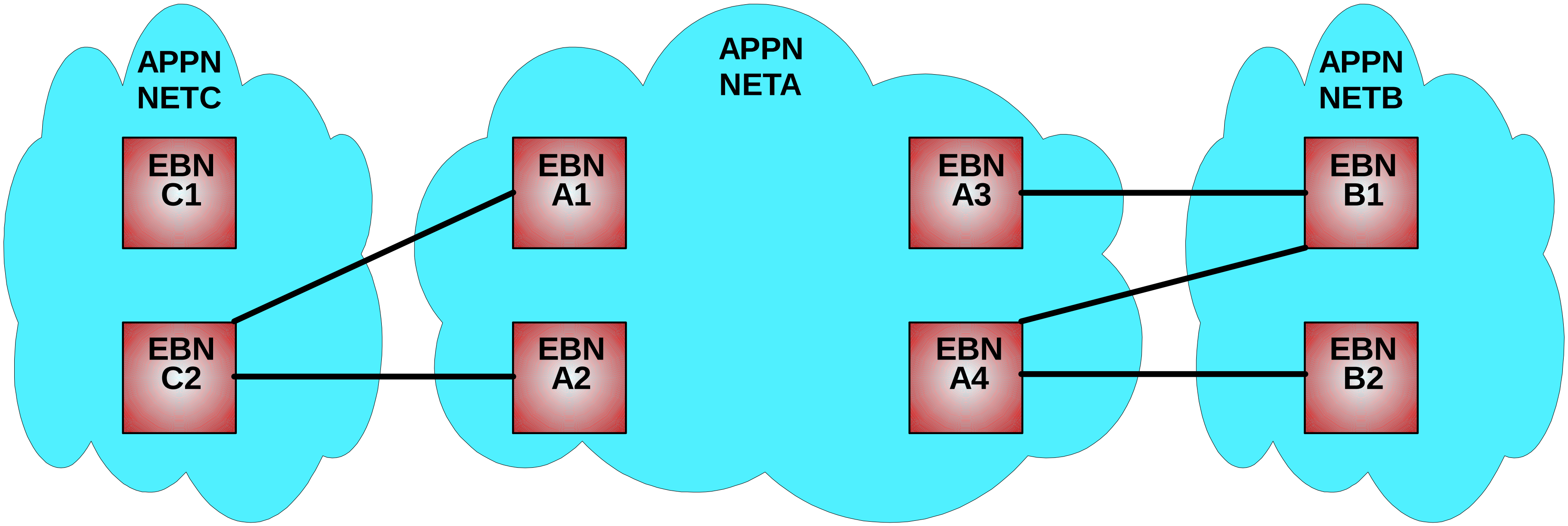
The target LU's NNS is the closest ***visible*** network node to the target LU's owning CP

Who is my NNS?



- **Who is the NNS for:**
  - **NETA.A3 ?**
  - **NETA.A7 ?**
  - **NETZ.Z9 ?**
  - **NETY.Y8 ?**
  - **NETA.A5 ?**
  - **NETA.A3**
  - **NETA.A2**
  - **NETA.A4**
  - **NETA.A1**
  - **NETA.A1 or NETA.A2**

What about cross-network searches?



- **Who do these nodes think is the NNS for NETC.C1?**

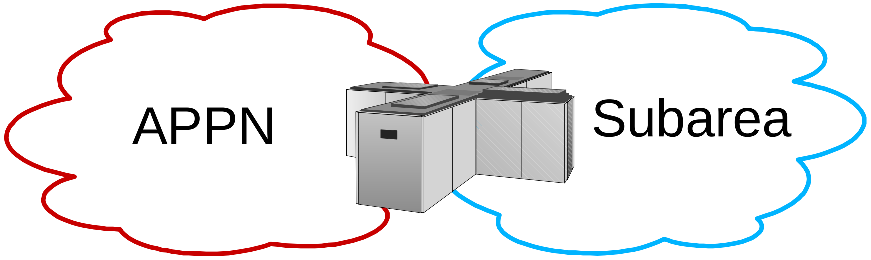
● **NETC.C2 ?**	● **NETC.C1**
● **NETA.A1 ?**	● **NETC.C2 or NETA.A2**
● **NETA.A2 ?**	● **NETC.C2 or NETA.A1**
● **NETA.A4 ?**	● **NETA.A1 or NETA.A2**
● **NETB.B1 ?**	● **NETA.A3 or NETA.A4 or NETB.B2**
● **NETB.B2 ?**	● **NETA.A4 or NETB.B1**

Agenda

- Searching using ALIAS names
- Who is my network node server (NNS)?
- Introduction to Subarea and APPN searching
- Tuning Subarea Searching
  - Subarea search controls
  - ADJSSCP tables for sample subarea-to-APPN migration
- Tuning APPN EBN Searching
  - Introduction to extended border nodes (EBNs)
  - APPN search controls
  - Subnetwork routing lists (SRLs)
  - ADJCLUST tables for sample mixed APPN and subarea network
- Hints to aid in debugging search problems

Subarea and APPN searching introduction

- VTAM has two distinct search algorithms: Subarea & APPN
  - Understanding details of either is difficult
  - Understanding details of both is very difficult
  - Understanding how they interact is almost impossible!



- Which search algorithm is used?
  - Depends on the origin of search (subarea or APPN)
  - Searching switches between subarea and APPN as needed
- Tuning for performance may require some work
  - Especially when parallel APPN and subarea paths exist (e.g. VRTGs)
    - Trade-Off: dynamics versus predefinition
  - Complexity is significantly reduced when network is completely APPN

Overview of subarea searching

- Trial-and-error routing using ADJSSCP tables

- Different tables allowed for different resources
  - NETID known, CDRM known, NETID and CDRM known
  - Default table is used when no matching table is found
  - ADJLISTs associate specific tables with CDRSCs

- SME ADJSSCP selection function allows re-ordering

- Enhanced to allow control over APPN searching
  - "ISTAPNCP" entry means "search APPN network"
    - Includes adjacent subnetwork (border node) searching
  - ADJSSCP tables may not require changes (depending on SORDER)

Overview of APPN searching

- Registration of local resources
  - ENs register to CDServer and/or NNS; NNs register to CDServer
- Searches may be directed or broadcast
  - Directed search when target location is known (RDTE, DS DB, TOPO DB)
  - Broadcast search of domain ENs and local subnetwork
  - Serial directed search of interchange nodes (ICNs) and adjacent subnetworks (EBNs)
- CDServers reduce frequency of broadcasts
  - CDServer becomes focal point for network broadcasts
  - Origin CDServer queries other CDServers before sending broadcast
- Adjacent subnetwork (EBN) search uses ADJCLUST tables
  - Different tables allowed for different networks
  - "This EBN" entry means "search local subnetwork"
- Directory services management exit (DSME)
  - Allows search steps to be eliminated
  - Supports adding, deleting and re-ordering ADJCLUST entries

Agenda

- Searching using ALIAS names
- Who is my network node server (NNS)?
- Introduction to Subarea and APPN searching
- Tuning Subarea Searching
  - Subarea search controls
  - ADJSSCP tables for sample subarea-to-APPN migration
- Tuning APPN EBN Searching
  - Introduction to extended border nodes (EBNs)
  - APPN search controls
  - Subnetwork routing lists (SRLs)
  - ADJCLUST tables for sample mixed APPN and subarea network
- Hints to aid in debugging search problems

Subarea search controls

- **ADJSSCP tables and ADJLISTs**
  - Define subarea search order by target NETID and/or owning CDRM, or by CDRSC
- DIALRTRY=YES | NO
  - Reroute search requests on dial-out failure
- DYNASSCP=YES | NO
  - Create dynamic default ADJSSCP table
- MAXSSCPS=10 | nn (1-255)
  - SSCP visit count (subarea searching)
- **SORDER=APPN | APPNFRST | SUBAREA | ADJSSCP**
  - Define subarea versus APPN search order
- SSCPDYN=YES | NO
  - Update history information of origin resource
- SSCPORD=PRIORITY | DEFINED
  - Priority (history) or defined searching

Choosing SORDER (search order)

SORDER=APPNFRST, APPN or SUBAREA

- "ISTAPNCP" automatically added to ADJSSCP tables
  - At the top, near the top or at the bottom (respectively)
  - SSCPORD=PRIORITY can affect where ISTAPNCP is added!
- Start option affects all ADJSSCP tables (all NETIDs)
  - Unless SORDER is coded on the ADJSSCP table
  - Set start option based on where most resources reside (APPN Or Subarea)
  - Override on ADJSSCP tables as needed

SORDER=ADJSSCP

- "ISTAPNCP" must be explicitly coded in ADJSSCP tables
- Start option allows customized placement of "ISTAPNCP" by table (NETID), but...
  - Coding SORDER on each ADJSSCP table is recommended
- Use for ADJSSCP tables that should never search APPN
  - Reminder: SORDER=SUBAREA still searches APPN!

SORDER and SSCPORD summary

	SORDER			
SSCPORD	APPNFRST	APPN	ADJSSCP	SUBAREA
PRIORITY	1. APPN network 2. Learned owner 3. Coded owner 4. Prev. successes 5. ADJSSCP table 6. Prev. failures	1. Learned owner 2. Coded owner 3. APPN DS DB 4. Prev. successes 5. APPN network 6. ADJSSCP table 7. Prev. failures	1. Learned owner 2. Coded owner 3. APPN DS DB 4. Prev. successes 5. ADJSSCP table 6. Prev. failures	1. Learned owner 2. Coded owner 3. APPN DS DB 4. Prev. successes 5. ADJSSCP table 6. Prev. failures 7. APPN network
DEFINED	1. APPN network 2. Learned owner 3. Coded owner 4. ADJSSCP table	1. Learned owner 2. Coded owner 3. APPN network 4. ADJSSCP table	1. Learned owner 2. Coded owner 3. APPN DS DB 4. ADJSSCP table	1. Learned owner 2. Coded owner 3. APPN DS DB 4. ADJSSCP table 5. APPN network

Prefers APPN ↔ Prefers Subarea

Notes:

1. SORDER and SSCPORD do not apply to CDRSCs with ADJLIST coded (but ADJLISTs can include ISTAPNCP entry).
2. SORDER and SSCPORD only affect "subarea searches" (originated on this VTAM or received from an adjacent CDRM), except:
3. If SSCPORD=DEFINED, "APPN DS DB" Search is limited to resources on served ENs; If SSCPORD=PRIORITY, any DS DB entry may be used.
4. ISTAPNCP is ignored in ADJSSCP tables, unless SORDER=ADJSSCP.
5. If SORDER=ADJSSCP, APPN DS DB and APPN network will only be searched if ISTAPNCP is coded in the selected ADJSSCP table.
6. "Previous successes" and "Previous failures" can include ISTAPNCP.

Adjacent SSCP lists (ADJLISTs)

ADJLISTs

- Define explicit ADJSSCP search lists (by name)
  - ADJLISTs are coded within ADJSSCP tables
  - ADJLIST name is coded on CDRSC or GROUP definitions
- When searching for a CDRSC with ADJLIST coded:
  - SORDER, SSCPORD, learned and coded owning CDRMs are ignored!
  - What you code in the ADJLIST is what you get!

Use ADJLISTs:

- When tight control over search order is desired for some predefined CDRSCs
- When a simple search strategy is adequate for most CDRSCs
  - Predefine CDRSCs with ADJLISTs for the exceptions

Recommended search strategy

Code SORDER= on ADJSSCP tables!

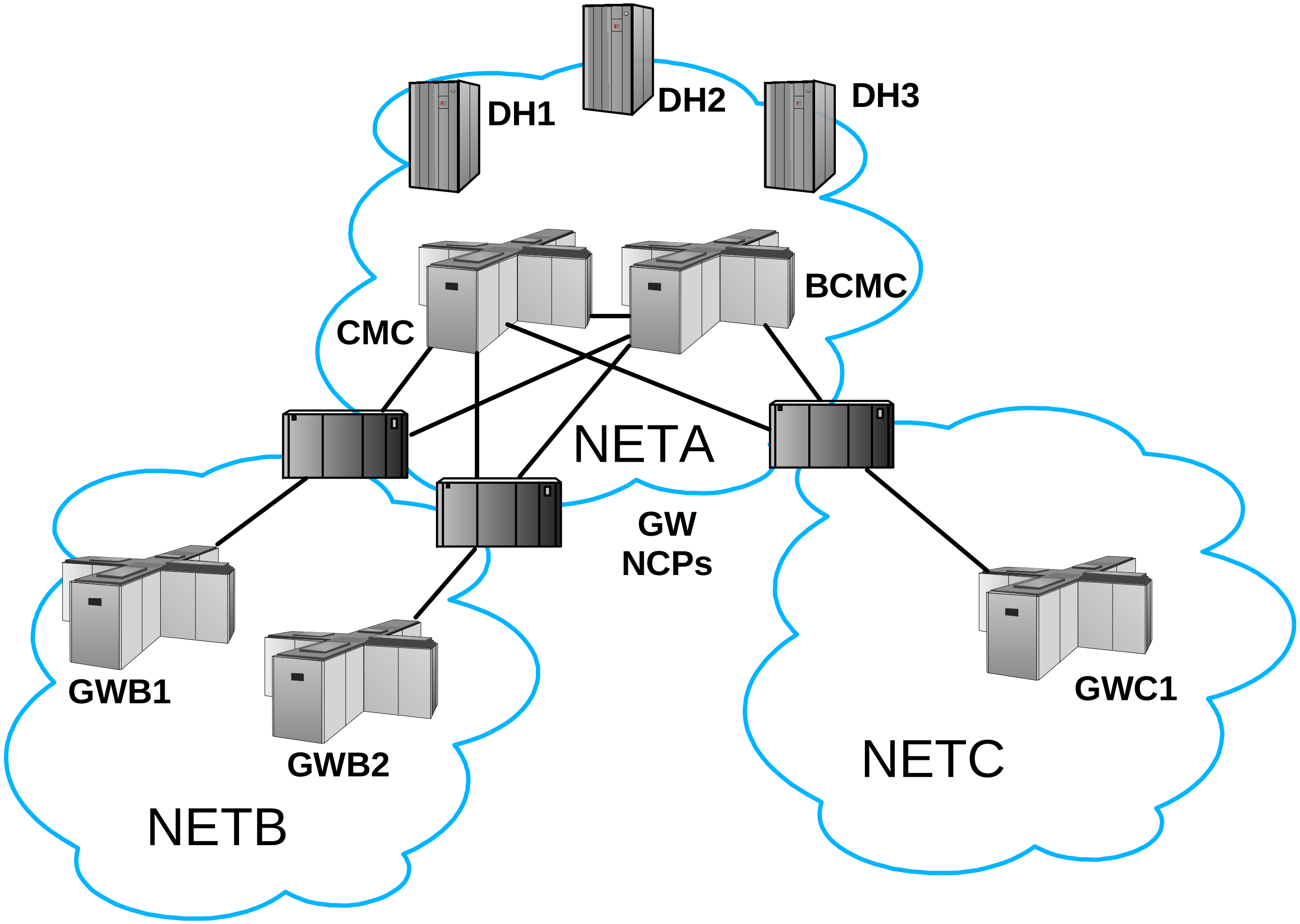
- Allows a different SORDER for each ADJSSCP table
  - Code SORDER= on NETWORK and/or CDRM statements
  - SORDER=STARTOPT is the default value for NETWORK statements
  - SORDER= value on NETWORK sifts down to subordinate CDRM tables
  - SORDER=STARTOPT on CDRM statement overrides SORDER value on preceding NETWORK statement with SORDER start option value
  - Note: MODIFY VTAMOPTS,SORDER= has an immediate affect on ADJSSCP tables with SORDER=STARTOPT coded or defaulted
- SORDER is not allowed on ADJLISTs (WYSIWYG)

Agenda

- Searching using ALIAS names
- Who is my network node server (NNS)?
- Introduction to Subarea and APPN searching
- Tuning Subarea Searching
  - Subarea search controls
  - ADJSSCP tables for sample subarea-to-APPN migration
- Tuning APPN EBN Searching
  - Introduction to extended border nodes (EBNs)
  - APPN search controls
  - Subnetwork routing lists (SRLs)
  - ADJCLUST tables for sample mixed APPN and subarea network
- Hints to aid in debugging search problems

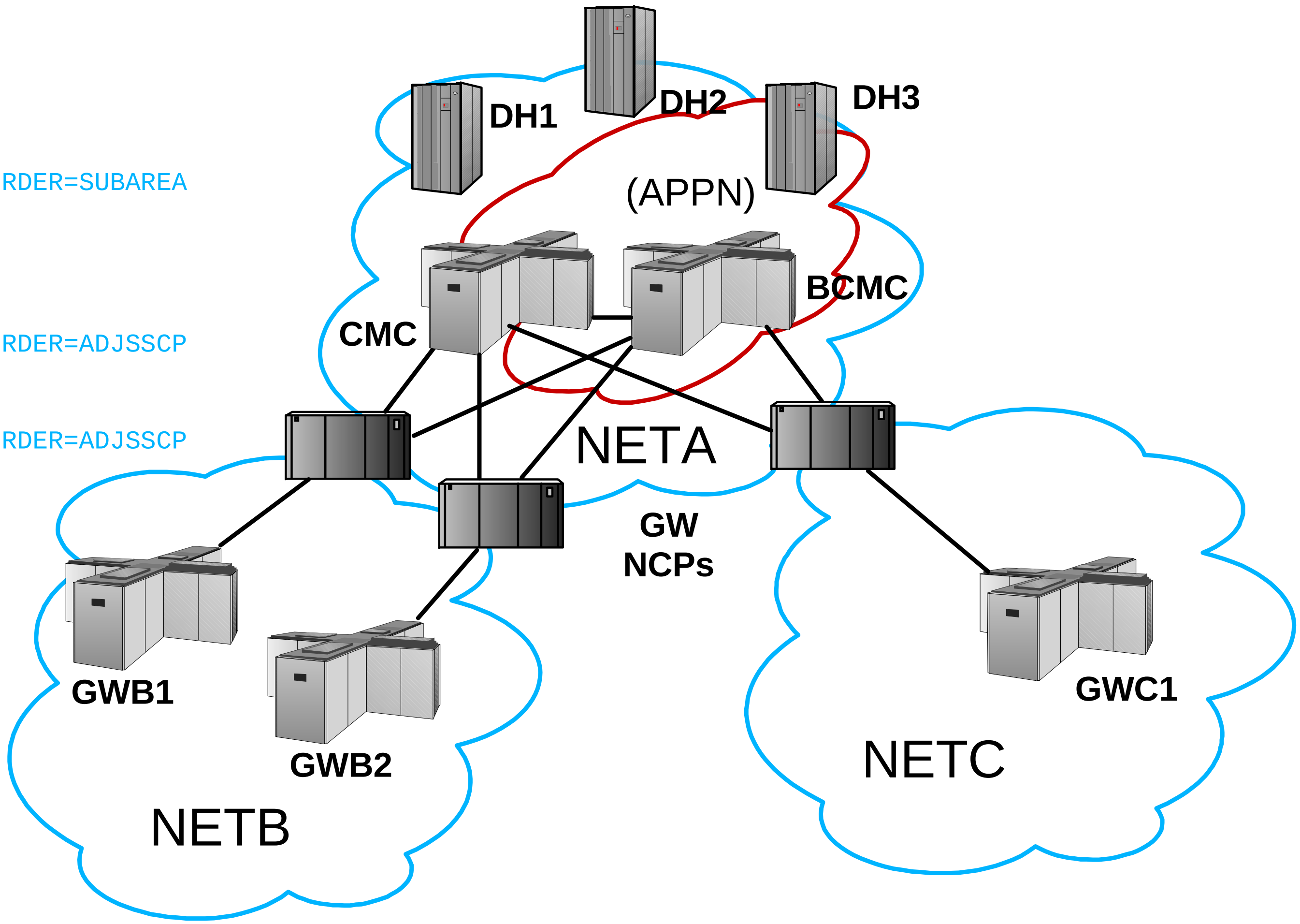
Sample network migration - all subarea

```
ADJSSCP  VBUILD  TYPE=ADJSSCP
DEFAULT  NETWORK
CMC      ADJCDRM
BCMC     ADJCDRM
DH1      ADJCDRM
DH2      ADJCDRM
DH3      ADJCDRM
GWB1     ADJCDRM
GWB2     ADJCDRM
GWC1     ADJCDRM
NETA     NETWORK NETID=NETA
CMC      ADJCDRM
BCMC     ADJCDRM
DH1      ADJCDRM
DH2      ADJCDRM
DH3      ADJCDRM
NETB     NETWORK NETID=NETB
GWB1     ADJCDRM
GWB2     ADJCDRM
NETC     NETWORK NETID=NETC
GWC1     ADJCDRM
```



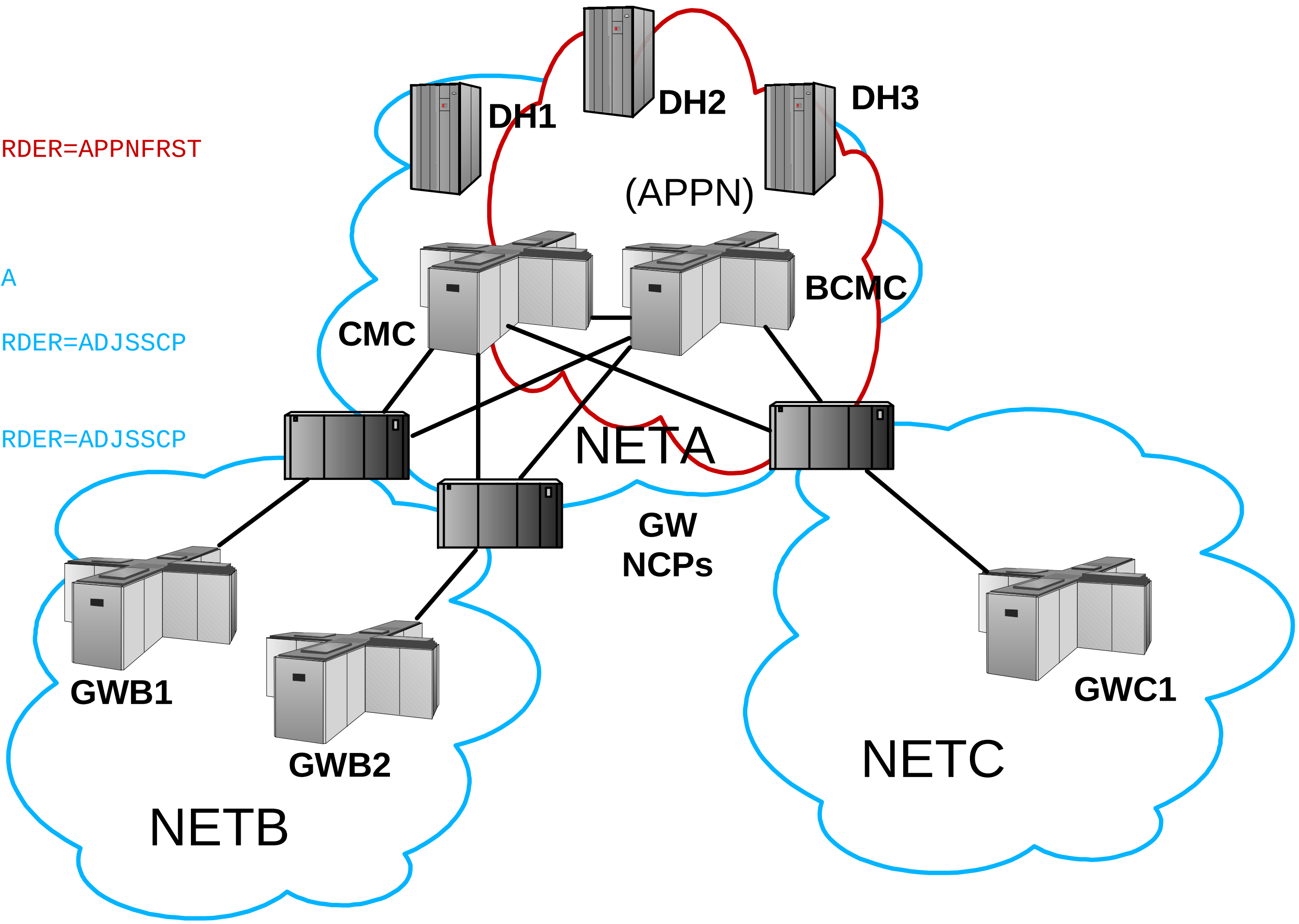
Sample network migration - mostly subarea

```
ADJSSCP   VBUILD  TYPE=ADJSSCP
DEFAULT   NETWORK SORDER=SUBAREA
CMC       ADJCDRM
BCMC      ADJCDRM
DH1       ADJCDRM
DH2       ADJCDRM
GWB1      ADJCDRM
GWB2      ADJCDRM
GWC1      ADJCDRM
NETA      NETWORK NETID=NETA,SORDER=SUBAREA
CMC       ADJCDRM
BCMC      ADJCDRM
DH1       ADJCDRM
DH2       ADJCDRM
NETB      NETWORK NETID=NETB,SORDER=ADJSSCP
GWB1      ADJCDRM
GWB2      ADJCDRM
NETC      NETWORK NETID=NETC,SORDER=ADJSSCP
GWC1      ADJCDRM
```



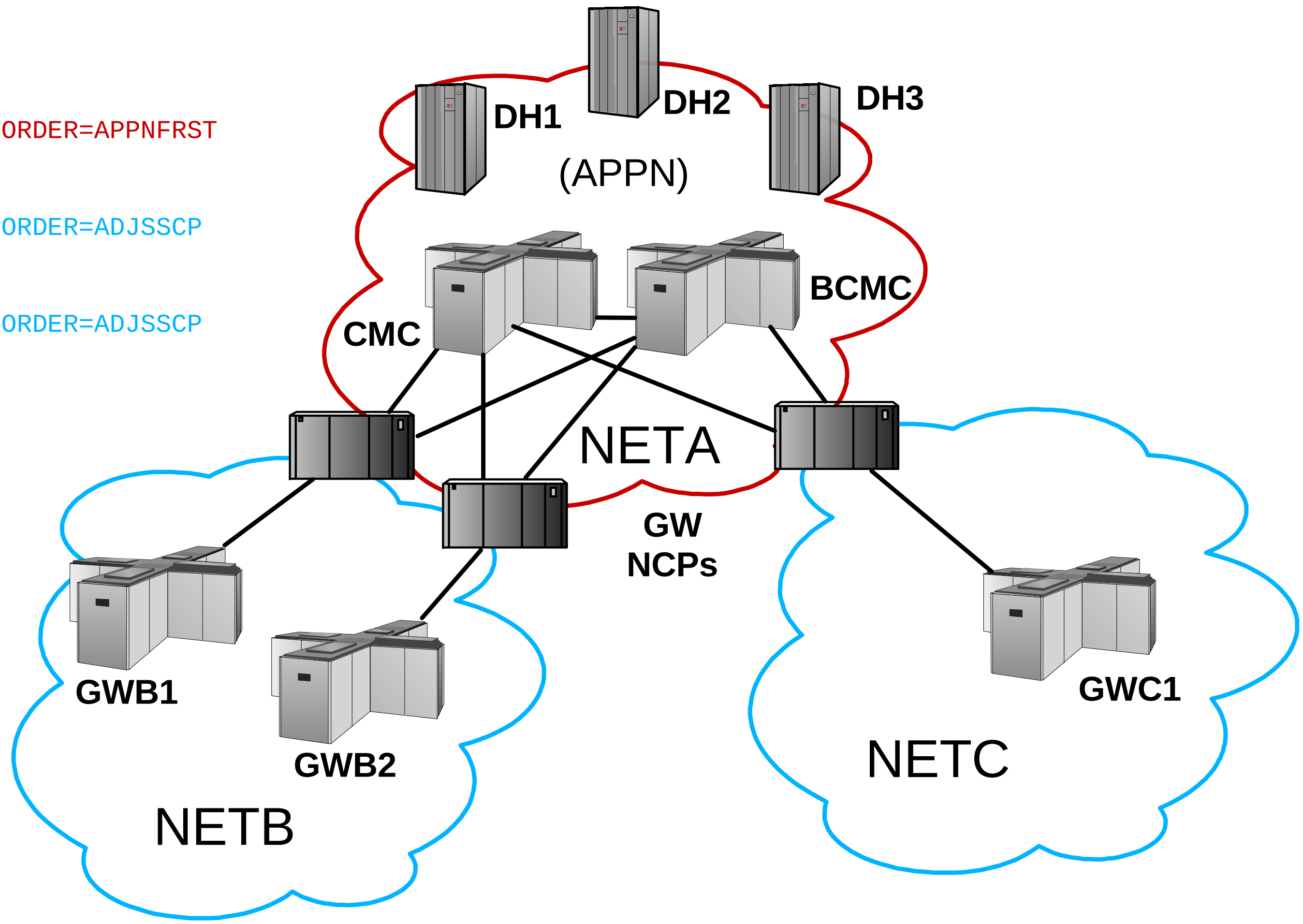
Sample network migration - mostly APPN

```
ADJSSCP  VBUILD  TYPE=ADJSSCP
DEFAULT  NETWORK SORDER=APPNFRST
CMC      ADJCDRM
BCMC     ADJCDRM
DH1      ADJCDRM
GWB1     ADJCDRM
GWB2     ADJCDRM
GWC1     ADJCDRM
NETA     NETWORK NETID=NETA,SORDER=APPNFRST
CMC      ADJCDRM
BCMC     ADJCDRM
DH1      ADJCDRM
DH1      CDRM    SORDER=SUBAREA
DH1      ADJCDRM
NETB     NETWORK NETID=NETB,SORDER=ADJSSCP
GWB1     ADJCDRM
GWB2     ADJCDRM
NETC     NETWORK NETID=NETC,SORDER=ADJSSCP
GWC1     ADJCDRM
```



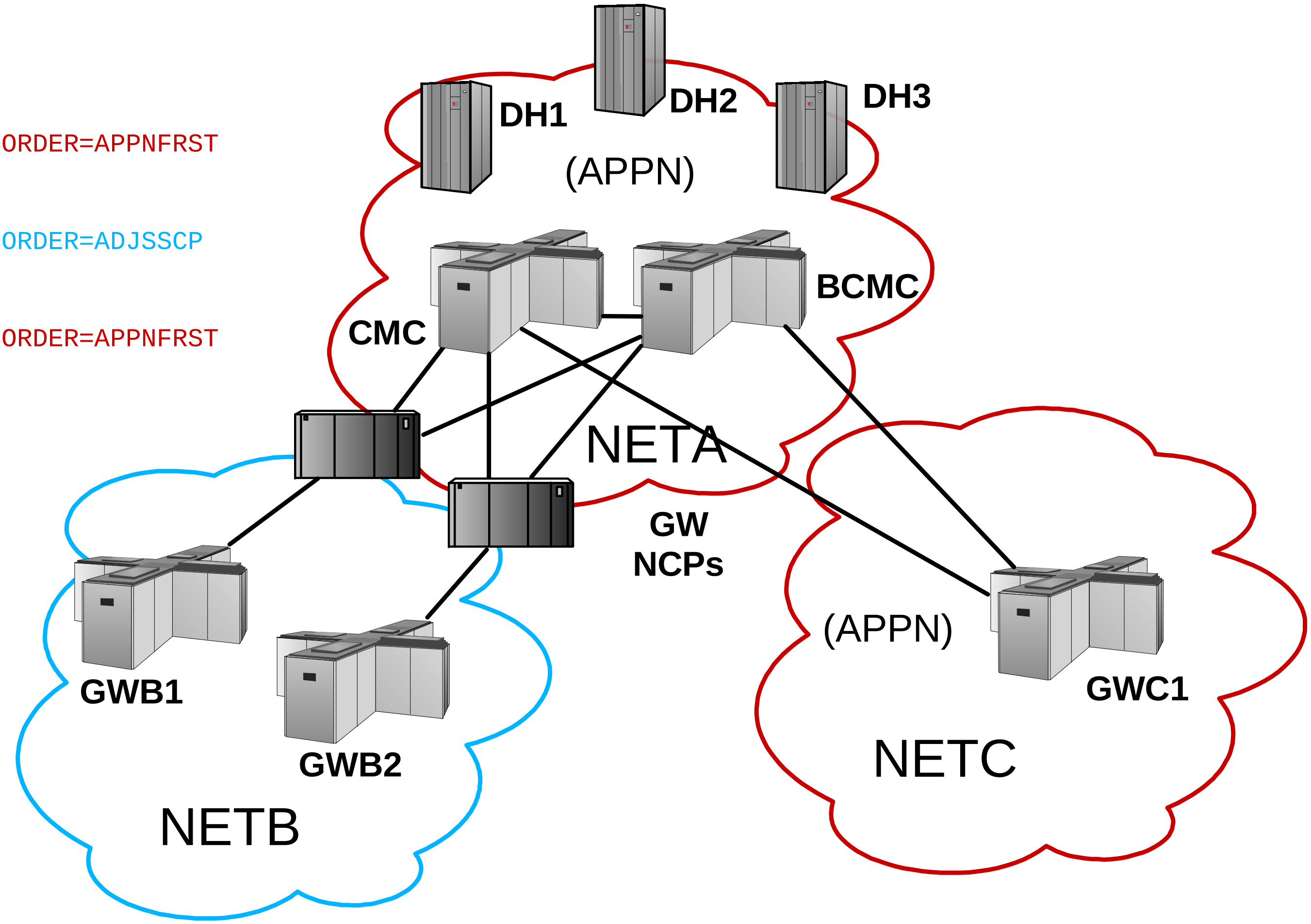
Sample network migration - all APPN

```
ADJSSCP   VBUILD   TYPE=ADJSSCP
DEFAULT   NETWORK SORDER=APPNFRST
CMC       ADJCDRM
BCMC      ADJCDRM
GWB1      ADJCDRM
GWB2      ADJCDRM
GWC1      ADJCDRM
NETA      NETWORK NETID=NETA,SORDER=APPNFRST
CMC       ADJCDRM
BCMC      ADJCDRM
NETB      NETWORK NETID=NETB,SORDER=ADJSSCP
GWB1      ADJCDRM
GWB2      ADJCDRM
NETC      NETWORK NETID=NETC,SORDER=ADJSSCP
GWC1      ADJCDRM
```



Sample network migration - SNI to EBN

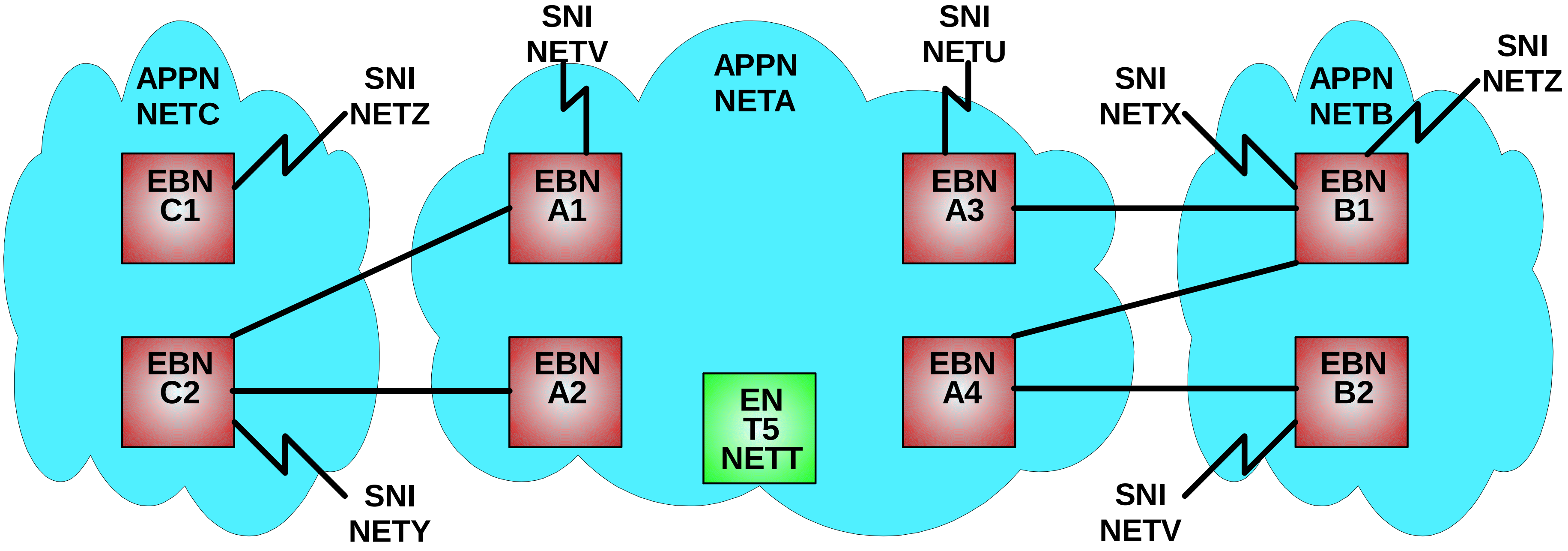
```
ADJSSCP   VBUILD  TYPE=ADJSSCP
DEFAULT   NETWORK SORDER=APPNFRST
CMC       ADJCDRM
BCMC      ADJCDRM
GWB1      ADJCDRM
GWB2      ADJCDRM
GWC1      ADJCDRM
NETA      NETWORK NETID=NETA,SORDER=APPNFRST
CMC       ADJCDRM
BCMC      ADJCDRM
NETB      NETWORK NETID=NETB,SORDER=ADJSSCP
GWB1      ADJCDRM
GWB2      ADJCDRM
NETC      NETWORK NETID=NETC,SORDER=APPNFRST
GWC1      ADJCDRM
```



Agenda

- Searching using ALIAS names
- Who is my network node server (NNS)?
- Introduction to Subarea and APPN searching
- Tuning Subarea Searching
  - Subarea search controls
  - ADJSSCP tables for sample subarea-to-APPN migration
- Tuning APPN EBN Searching
  - Introduction to extended border nodes (EBNs)
  - APPN search controls
  - Subnetwork routing lists (SRLs)
  - ADJCLUST tables for sample mixed APPN and subarea network
- Hints to aid in debugging search problems

Roles played by extended border nodes (EBNs)



- **Entry border node**
  - Receives a search request from an adjacent non-native APPN node
  - In the native subnetwork, acts as NNS(OLU) for non-native resources
- **Exit border node**
  - Sends a search request to an adjacent non-native APPN node
  - In the native subnetwork, acts as NNS(DLU) for non-native resources
- **Origin border node**
  - In OLU's subnetwork, first border node given control over searching
- **Peer border nodes**
  - Border nodes residing in the same subnetwork (native to each other) and each having connectivity to other adjacent networks

Little known facts about EBNs

- Peer EBN searches
  - EBN sends an EBN search to another native EBN
    - Allows the receiving EBN to search its adjacent subnetworks
  - Receiving EBN cannot reroute the EBN search to other peer EBNs
    - Peer EBNs are removed from the subnetwork routing list (SRL)
- When is an EBN not an EBN?   (No active EBN links!)
  - EBNs are defined using BN=YES start option, **but ...**
  - EBNs do not reported themselves as EBNs to the topology database (Topo DB) unless they have an active "intersubnetwork link" (ISL)
    - Other EBNs are not aware of peer EBNs until this occurs
  - EBNs remove native nodes that are not EBNs (in Topo DB) from SRLs
- When does an EBN search like an EBN? (Always!)
  - Even if no intersubnetwork links (ISLs) are active!
  - Allows EBN search controls to be used regardless of ISL status

Agenda

- Searching using ALIAS names
- Who is my network node server (NNS)?
- Introduction to Subarea and APPN searching
- Tuning Subarea Searching
  - Subarea search controls
  - ADJSSCP tables for sample subarea-to-APPN migration
- Tuning APPN EBN Searching
  - Introduction to extended border nodes (EBNs)
  - APPN search controls
  - Subnetwork routing lists (SRLs)
  - ADJCLUST tables for sample mixed APPN and subarea network
- Hints to aid in debugging search problems

APPN search controls

- **ADJCLUST tables**
  - Define EBN search order by target NETID
- **ALIASRCH=** (V1R8) and **AUTHNETS=** (V1R10)
  - ADJCP operands to prevent unwanted inbound non-native searches
- **BNDYN=LIMITED | FULL | NONE**
  - Border node search dynamics
- BNORD=PRIORITY | DEFINED
  - Priority (history) or defined EBN searching
- DUPDEFS=ALL | NONE | APPL | DEPLU
  - Duplicate APPL and/or LU definitions exist
- MAXLOCAT=5000 | nnnnn (threshold)
  - Locate search congestion control
- **SNVC=3 | nn** (1-255)
  - APPN subnetwork visit count (EBN searching)
- SSEARCH=YES | NO | CACHE | APPNFRST
  - Subarea search allowed

Border node dynamics (BNDYN)

- **BNDYN=FULL**
  - Adds all EBNs (and adjacent non-native NNs) to the SRL
    - No ADJCLUST tables are required, but performance is not optimal
- **BNDYN=NONE**
  - Never adds any EBNs (or adjacent non-native NNs) to the SRL
    - ADJCLUST tables must be defined for every NETID or the default table is used
- **BNDYN=LIMITED**
  - "Intelligently" adds EBNs (and adjacent non-native NNs) to the SRL
    - Nodes with the same NETID as the target LU
    - Nodes through which resources with target LU's NETID have been found
    - If target LU's NETID is not known (ALIAS search), then all nodes are searched
  - ADJCLUST tables still may be needed to:
    - Find resources through intermediate subnetworks or through peer EBNs
    - Find 'native' resources with different NETIDs (NNNA, LEN, EN, SNI, etc.)
- **BNDYN=** on ADJCLUST tables will override start option value
  - z/OS CS V1R9: BNORD= can also be coded on ADJCLUST tables

Subnetwork visit count (SNVC)

SNVC start option

- Number of subnetworks that can be on the search path (by default)
  - Includes the OLU's subnetwork (number of subnet boundaries + 1)
- Default SNVC start option value is 3
- SNVC on ADJCLUST tables will override start option value

SNVC operand in ADJCLUST tables

- On NETWORK statement
  - Default value used for all NEXTCPs in that ADJCLUST table
  - If not specified, defaults to SNVC start option value
- On NEXTCP statement
  - SNVC value used only when sending an EBN search to the specified CP
  - If not specified, defaults to SNVC value on preceding NETWORK statement

Adjacent cluster (ADJCLUST) tables

```
ADJCLUST VBUILD    TYPE=ADJCLUST
**********************************************************************
* Sample default ADJCLUST table for NETA.A4                          *
**********************************************************************
DEFAULT  NETWORK   BNDYN=NONE         dynamic entries not allowed
A4       NEXTCP    CPNAME=NETA.A4     "my name" (native subnet srch)
B1       NEXTCP    CPNAME=NETB.B1     adjacent non-native EBN
B2       NEXTCP    CPNAME=NETB.B2     adjacent non-native EBN
A1       NEXTCP    CPNAME=NETA.A1     native (peer) EBN
A2       NEXTCP    CPNAME=NETA.A2     native (peer) EBN
A3       NEXTCP    CPNAME=NETA.A3     native (peer) EBN
```

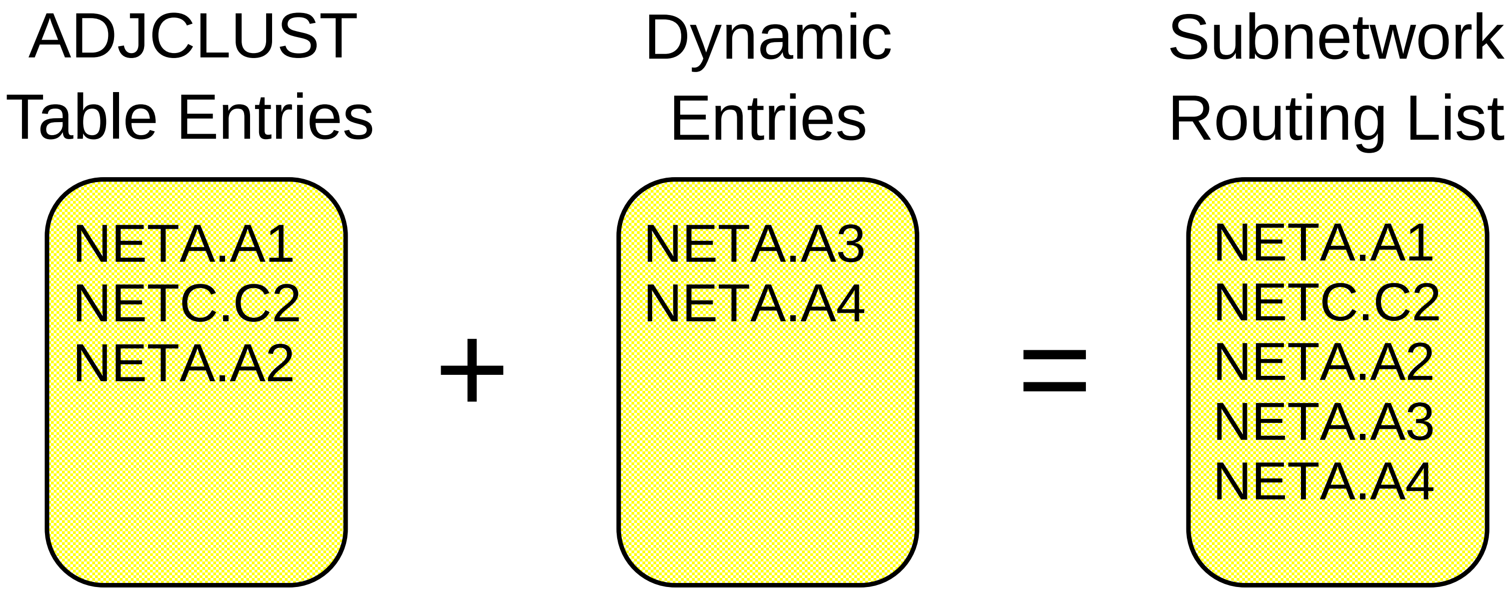
- What nodes can be defined in **ADJCLUST** tables?
  - This border node ("my name")
    - Initiates a search of the entire native subnetwork
    - Includes ENs and ICNs (SNI searches), regardless of NETID
  - Adjacent non-native border nodes and/or network nodes
    - Initiates a search of the adjacent subnetwork through that non-native node
    - Adjacent subnetwork can also search cross-subnetwork (if SNVC allows)
  - Peer border nodes
    - Initiates a search of subnetworks adjacent to other native EBNs

Agenda

- Searching using ALIAS names
- Who is my network node server (NNS)?
- Introduction to Subarea and APPN searching
- Tuning Subarea Searching
  - Subarea search controls
  - ADJSSCP tables for sample subarea-to-APPN migration
- Tuning APPN EBN Searching
  - Introduction to extended border nodes (EBNs)
  - APPN search controls
  - Subnetwork routing lists (SRLs)
  - ADJCLUST tables for sample mixed APPN and subarea network
- Hints to aid in debugging search problems

Subnetwork routing lists (SRLs)

- New subnetwork routing list (SRL) is created for each search
  - Contains complete list of nodes to search during cross-subnetwork (EBN) searching
  - Includes all entries from selected ADJCLUST table, as well as dynamic entries
    - Dynamic entries are learned from Topo DB, adjacency, and prior search history



- How many dynamic entries are added depends on BNDYN value

Subnetwork routing lists and BNDYN

- Assume that no ADJCLUST tables have been defined
  - The following SRLs would be used by NETA.A4 for the BNDYN values shown

	ADJCLUST		Dynamic		SRL
BNDYN=NONE	(none)	+	(none)	=	(none)
BNDYN=LIMITED	(none)	+	depends on NETID of target resource	=	depends on NETID of target resource
BNDYN=FULL	(none)	+	NETA.A1 NETA.A2 NETA.A3 NETA.A4 NETB.B1 NETB.B2	=	NETA.A1 NETA.A2 NETA.A3 NETA.A4 NETB.B1 NETB.B2

Subnetwork routing lists and BNDYN=LIMITED

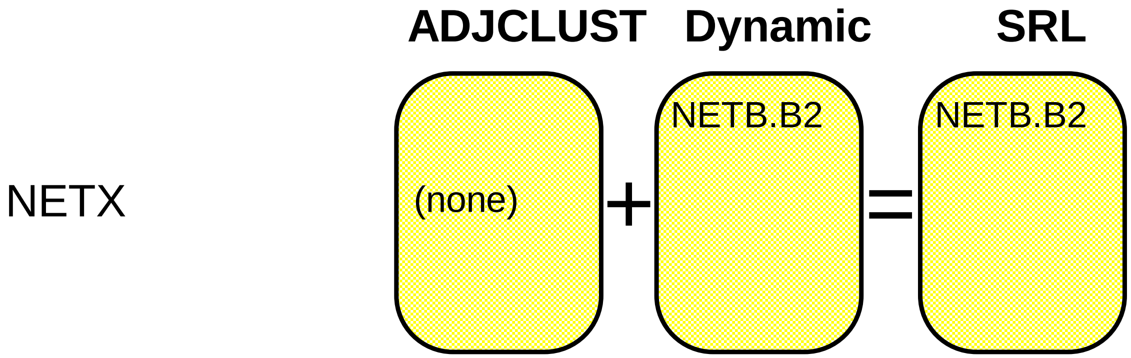
- Assume that no ADJCLUST tables have been defined
  - The following SRLs would be used by NETA.A4 for the NETIDs shown

	ADJCLUST		Dynamic		SRL
NETA	(none)	+	NETA.A1 NETA.A2 NETA.A3 NETA.A4	=	NETA.A1 NETA.A2 NETA.A3 NETA.A4
NETB	(none)	+	NETB.B1 NETB.B2	=	NETB.B1 NETB.B2
NETC	(none)	+	(none)	=	(none)

	ADJCLUST		Dynamic		SRL
NETT	(none)	+	(none)	=	(none)
NETX	(none)	+	(none)	=	(none)
alias	(none)	+	NETA.A1 NETA.A2 NETA.A3 NETA.A4 NETB.B1 NETB.B2	=	NETA.A1 NETA.A2 NETA.A3 NETA.A4 NETB.B1 NETB.B2

SRLs, BNDYN=LIMITED and prior search history

- Assume that no ADJCLUST tables have been defined
- Assume a NETX resource sends a search into NETA through B2
  - NETA.A4 has now learned that NETX resources can be found through NETB
  - The following SRL would then be used by NETA.A4 for NETX searches



	ADJCLUST		Dynamic		SRL
NETX	(none)	+	NETB.B2	=	NETB.B2

  - Notice that **NOT ALL** EBNs in NETB are added to the SRL for NETX searches! (Only the EBN through which the search from NETX was received.)

Summary of possible BNDYN options

BNDYN=FULL

- Works all of the time
- Searching is not optimized at all
  - Can result in a lot unnecessary searching (CPU and network overhead)
  - Adjacent subnetworks (including SNI) could see unnecessary searches too!

BNDYN=LIMITED

- Most "intelligent" searching option available
- Works only for networks with very simple network interconnectivity
  - All resources must reside in immediately adjacent APPN subnetworks
  - No native resources with different NETIDs (SNI, LEN, EN or NNNA)

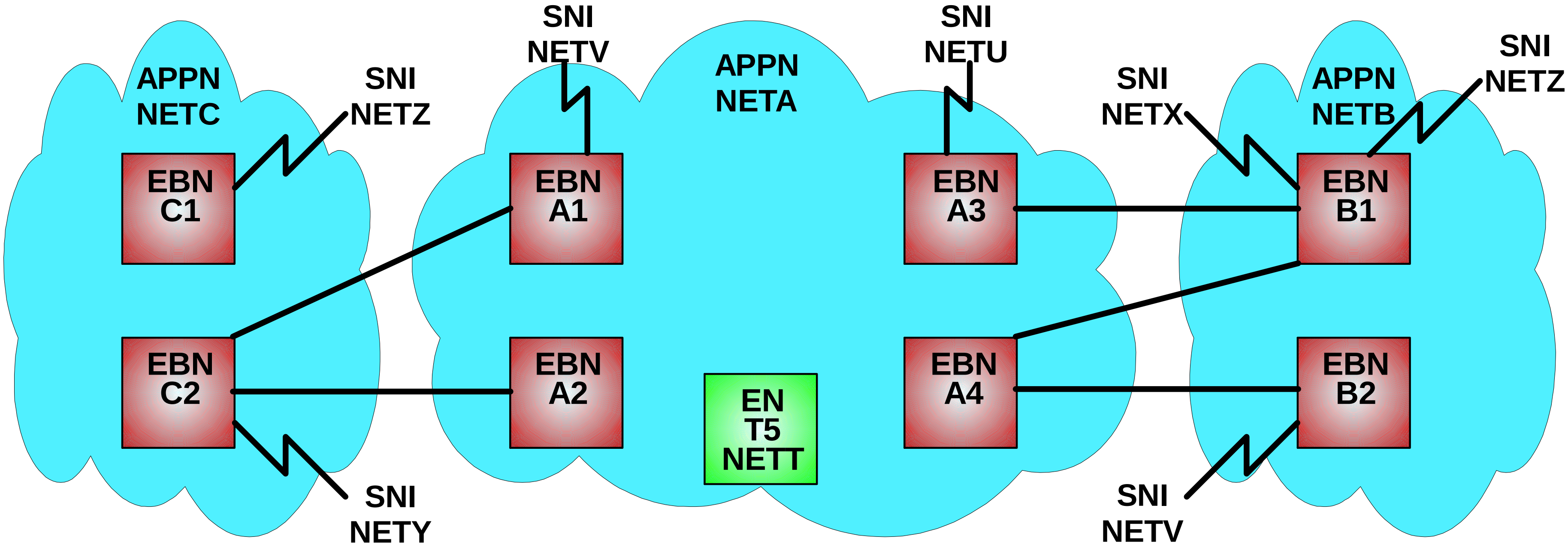
BNDYN=NONE

- Least "intelligent" searching option available
- Requires ADJCLUST tables for every possible target NETID (plus default table)
- **Will probably be needed by most customers who want optimal searching**
  - Some customers may use BNDYN=LIMITED and predefine additional entries

Agenda

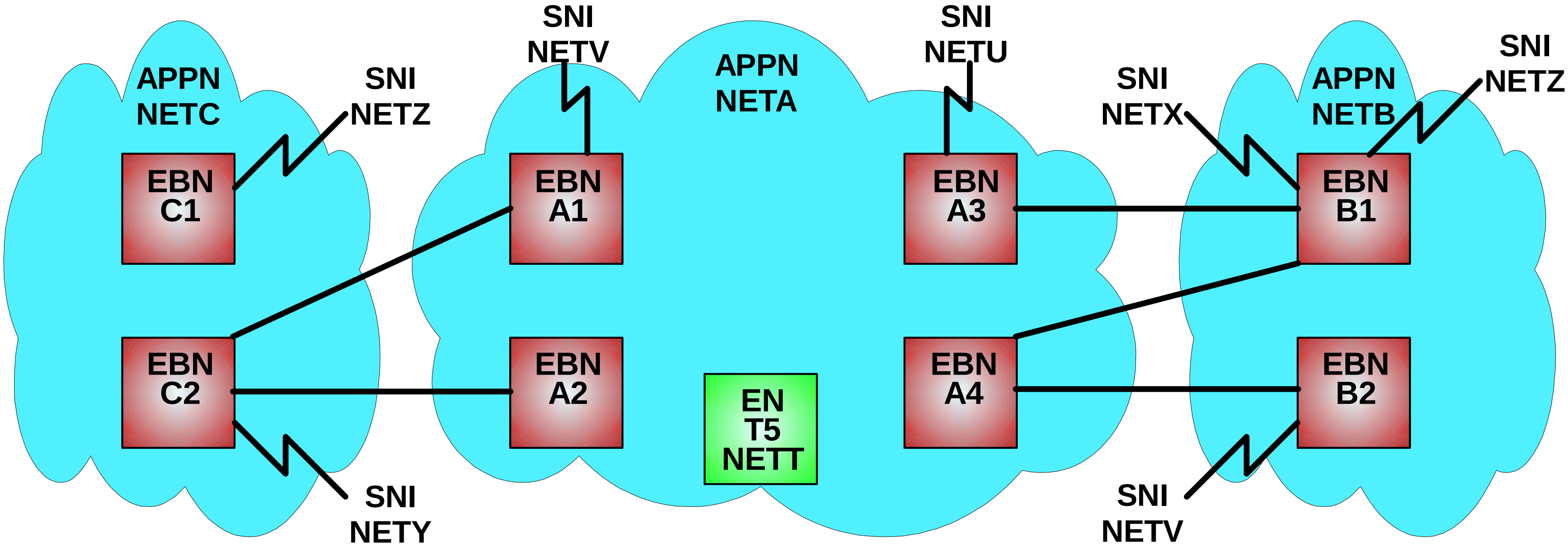
- Searching using ALIAS names
- Who is my network node server (NNS)?
- Introduction to Subarea and APPN searching
- Tuning Subarea Searching
  - Subarea search controls
  - ADJSSCP tables for sample subarea-to-APPN migration
- Tuning APPN EBN Searching
  - Introduction to extended border nodes (EBNs)
  - APPN search controls
  - Subnetwork routing lists (SRLs)
  - ADJCLUST tables for sample mixed APPN and subarea network
- Hints to aid in debugging search problems

NETA.A1 ADJCLUST Tables (BNDYN=NONE) (a)



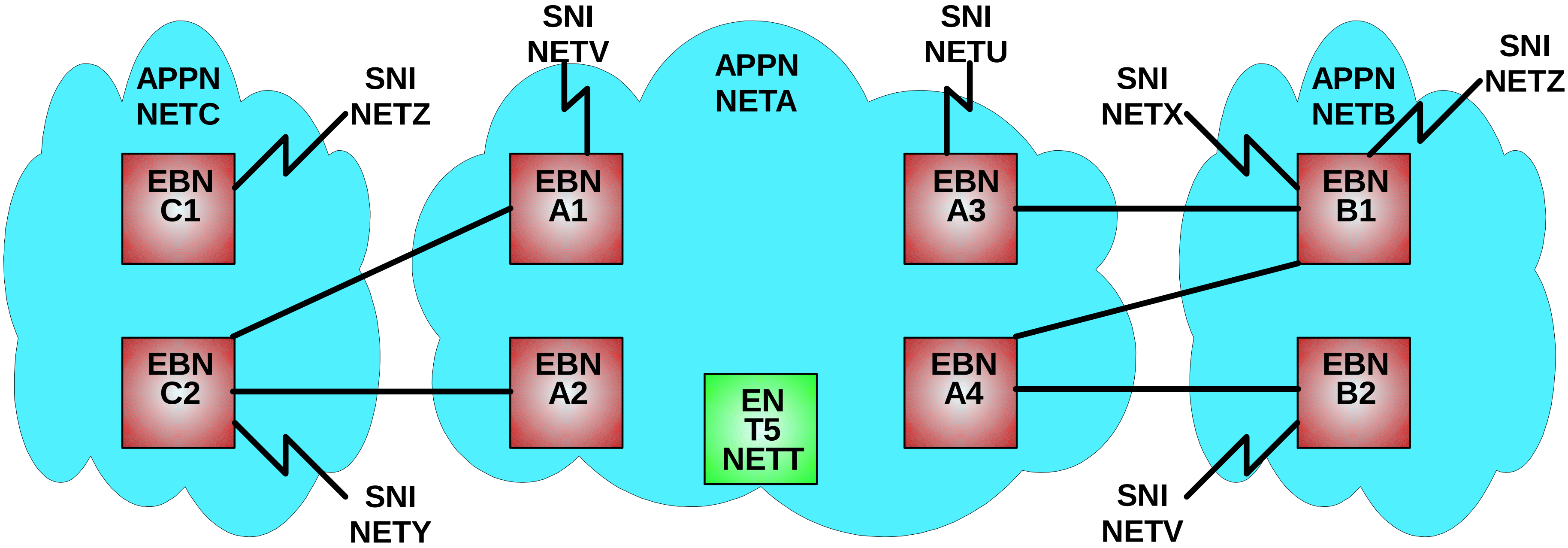
```
ADJCLUST VBUILD    TYPE=ADJCLUST
NETA     NETWORK   NETID=NETA
A1       NEXTCP    CPNAME=NETA.A1
*
NETT     NETWORK   NETID=NETT
A1       NEXTCP    CPNAME=NETA.A1
*
NETU     NETWORK   NETID=NETU
A1       NEXTCP    CPNAME=NETA.A1
```

NETA.A1 ADJCLUST Tables (BNDYN=NONE) (b)



```
ADJCLUST VBUILD    TYPE=ADJCLUST
NATIVE   NETWORK   NETID=(NETA,NETT,   *
                          NETU)
A1       NEXTCP    CPNAME=NETA.A1
*
NETC     NETWORK   NETID=NETC
C2       NEXTCP    CPNAME=NETC.C2
A2       NEXTCP    CPNAME=NETA.A2
*
NETY     NETWORK   NETID=NETY
C2       NEXTCP    CPNAME=NETC.C2
A2       NEXTCP    CPNAME=NETA.A2
```

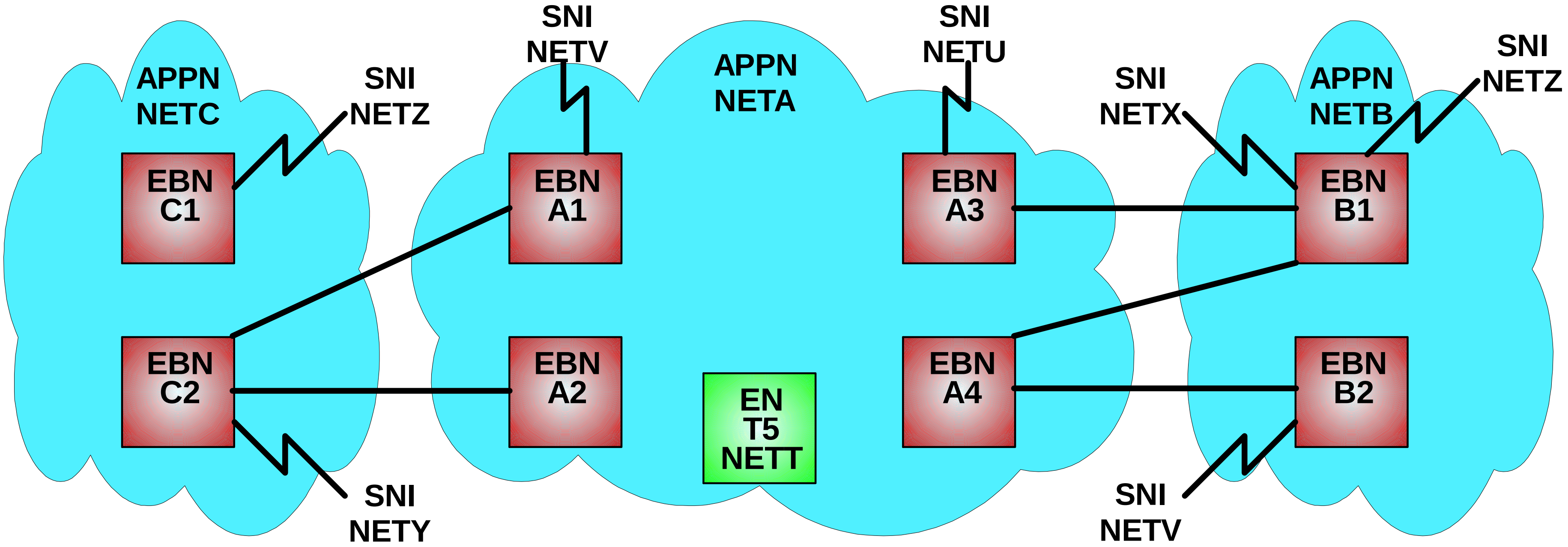
NETA.A1 ADJCLUST Tables (BNDYN=NONE) (c)



```
ADJCLUST VBUILD     TYPE=ADJCLUST
NATIVE   NETWORK    NETID=(NETA,NETT,           *
                           NETU)
A1       NEXTCP     CPNAME=NETA.A1
*
THRUNETC NETWORK    NETID=(NETC,NETY)
C2       NEXTCP     CPNAME=NETC.C2
A2       NEXTCP     CPNAME=NETA.A2
```

```
*
NETB     NETWORK    NETID=NETB
A3       NEXTCP     CPNAME=NETA.A3
A4       NEXTCP     CPNAME=NETA.A4
*
NETX     NETWORK    NETID=NETX
A3       NEXTCP     CPNAME=NETA.A3
A4       NEXTCP     CPNAME=NETA.A4
```

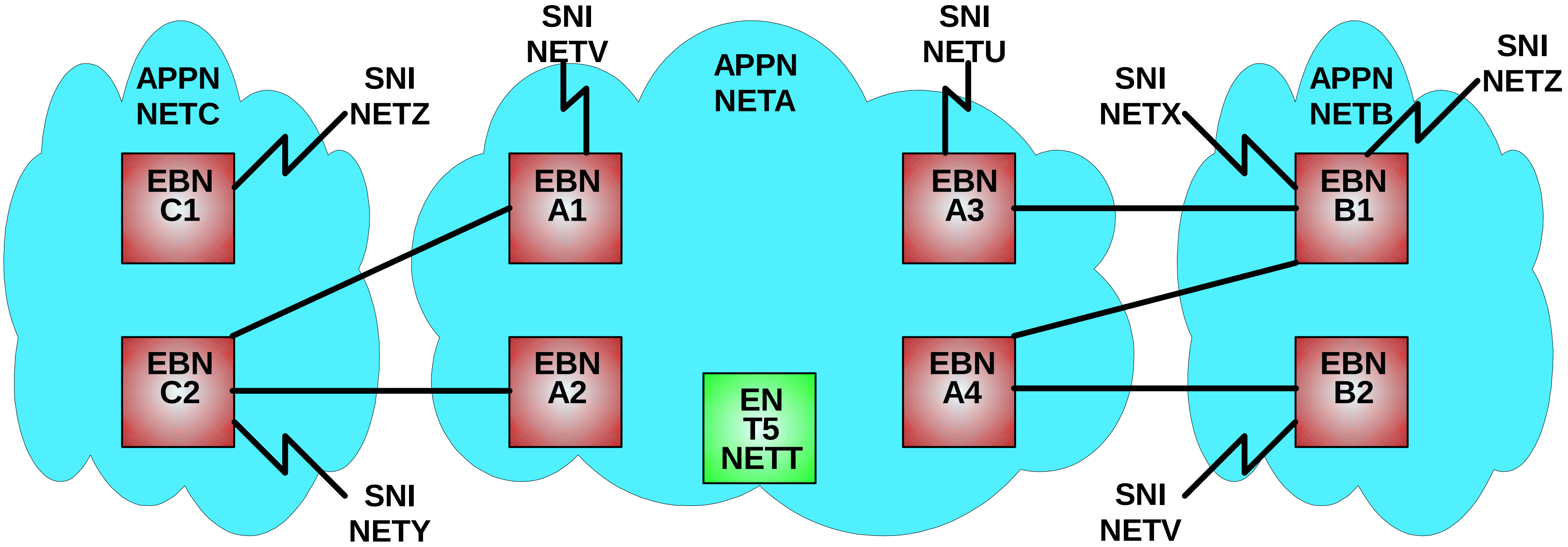
NETA.A1 ADJCLUST Tables (BNDYN=NONE) (d)



```
ADJCLUST VBUILD    TYPE=ADJCLUST
NATIVE   NETWORK   NETID=(NETA,NETT,      *
                          NETU)
A1       NEXTCP    CPNAME=NETA.A1
*
THRUNETC NETWORK   NETID=(NETC,NETY)
C2       NEXTCP    CPNAME=NETC.C2
A2       NEXTCP    CPNAME=NETA.A2
*
THRUNETB NETWORK   NETID=(NETB,NETX)
A3       NEXTCP    CPNAME=NETA.A3
A4       NEXTCP    CPNAME=NETA.A4
```

```
*
NETV     NETWORK   NETID=NETV
A1       NEXTCP    CPNAME=NETA.A1
A3       NEXTCP    CPNAME=NETA.A3
A4       NEXTCP    CPNAME=NETA.A4
*
NETZ     NETWORK   NETID=NETZ
C2       NEXTCP    CPNAME=NETC.C2
A2       NEXTCP    CPNAME=NETA.A2
A3       NEXTCP    CPNAME=NETA.A3
A4       NEXTCP    CPNAME=NETA.A4
```

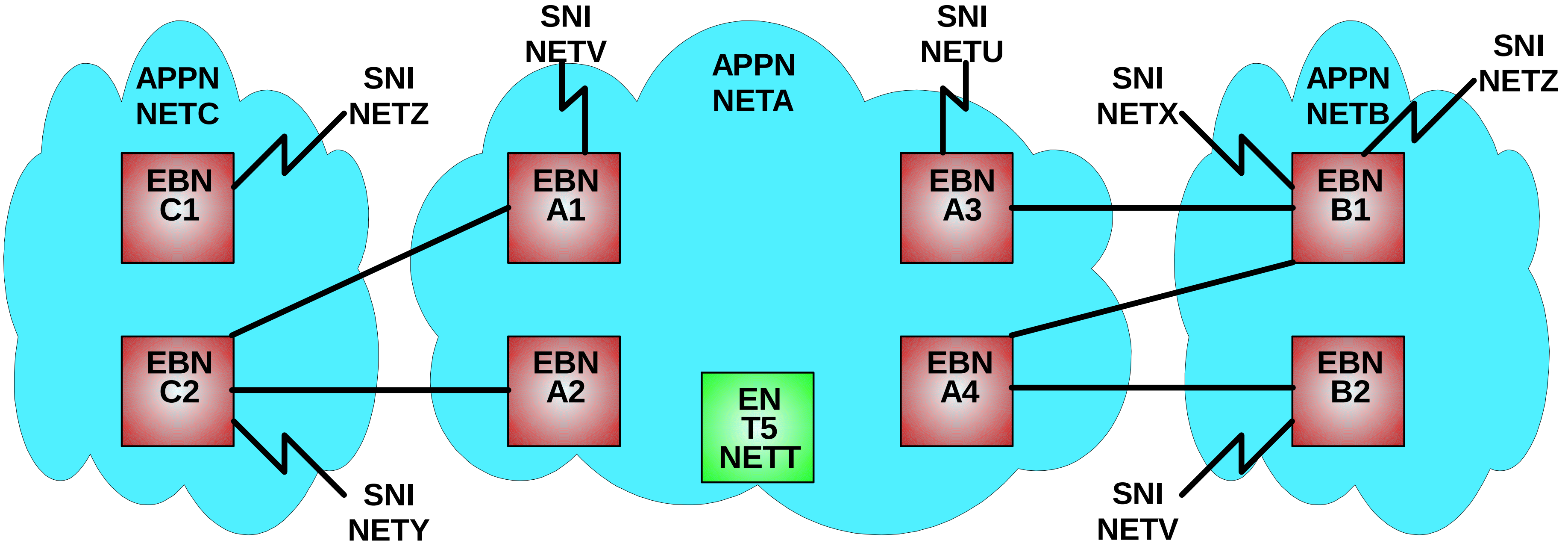
NETA.A2 ADJCLUST Tables (BNDYN=NONE)



```
ADJCLUST VBUILD   TYPE=ADJCLUST
NATIVE   NETWORK  NETID=(NETA,NETT,    *
                         NETU)
A2       NEXTCP   CPNAME=NETA.A2
*
THRUNETC NETWORK  NETID=(NETC,NETY)
C2       NEXTCP   CPNAME=NETC.C2
A1       NEXTCP   CPNAME=NETA.A1
*
THRUNETB NETWORK  NETID=(NETB,NETX)
A3       NEXTCP   CPNAME=NETA.A3
A4       NEXTCP   CPNAME=NETA.A4
```

```
*
NETV     NETWORK  NETID=NETV
A2       NEXTCP   CPNAME=NETA.A2
A3       NEXTCP   CPNAME=NETA.A3
A4       NEXTCP   CPNAME=NETA.A4
*
NETZ     NETWORK  NETID=NETZ
C2       NEXTCP   CPNAME=NETC.C2
A1       NEXTCP   CPNAME=NETA.A1
A3       NEXTCP   CPNAME=NETA.A3
A4       NEXTCP   CPNAME=NETA.A4
```

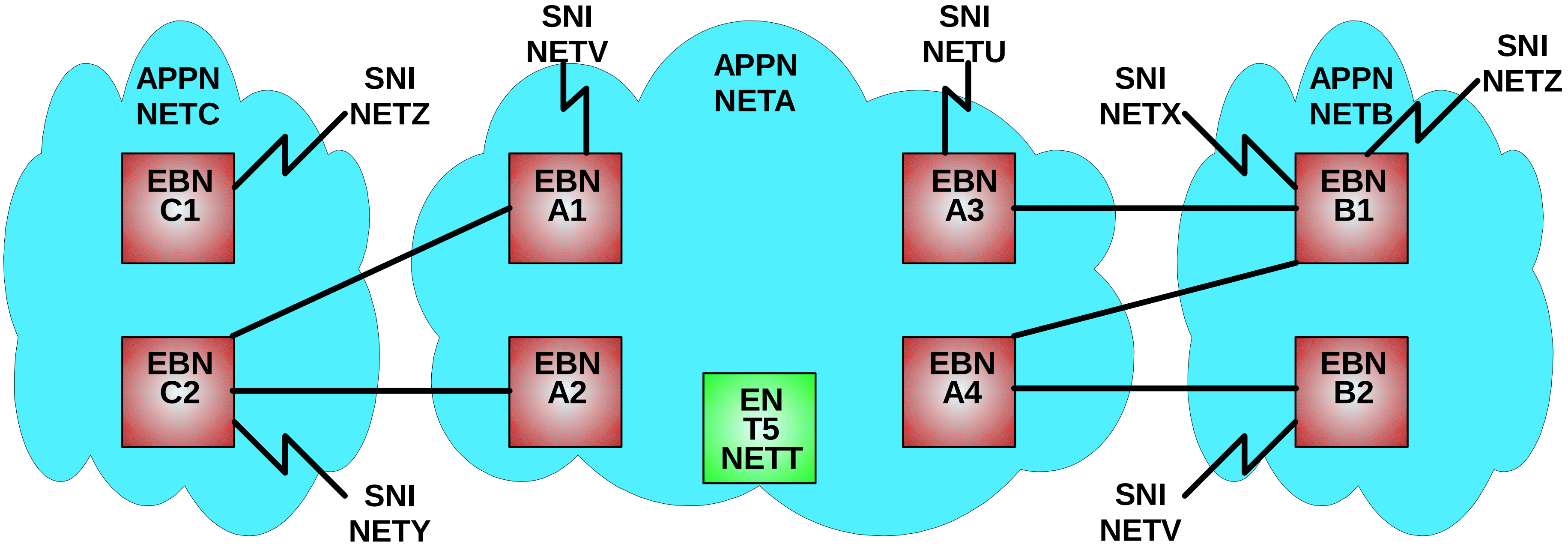
NETA.A3 ADJCLUST Tables (BNDYN=NONE)



```
ADJCLUST VBUILD   TYPE=ADJCLUST
NATIVE   NETWORK  NETID=(NETA,NETT,     *
                         NETU)
A3       NEXTCP   CPNAME=NETA.A3
*
THRUNETC NETWORK  NETID=(NETC,NETY)
A1       NEXTCP   CPNAME=NETA.A1
A2       NEXTCP   CPNAME=NETA.A2
*
THRUNETB NETWORK  NETID=(NETB,NETX)
B1       NEXTCP   CPNAME=NETB.B1
A4       NEXTCP   CPNAME=NETA.A4
```

```
*
NETV     NETWORK  NETID=NETV
A3       NEXTCP   CPNAME=NETA.A3
B1       NEXTCP   CPNAME=NETB.B1
A4       NEXTCP   CPNAME=NETA.A4
*
NETZ     NETWORK  NETID=NETZ
B1       NEXTCP   CPNAME=NETB.B1
A4       NEXTCP   CPNAME=NETA.A4
A1       NEXTCP   CPNAME=NETA.A1
A2       NEXTCP   CPNAME=NETA.A2
```

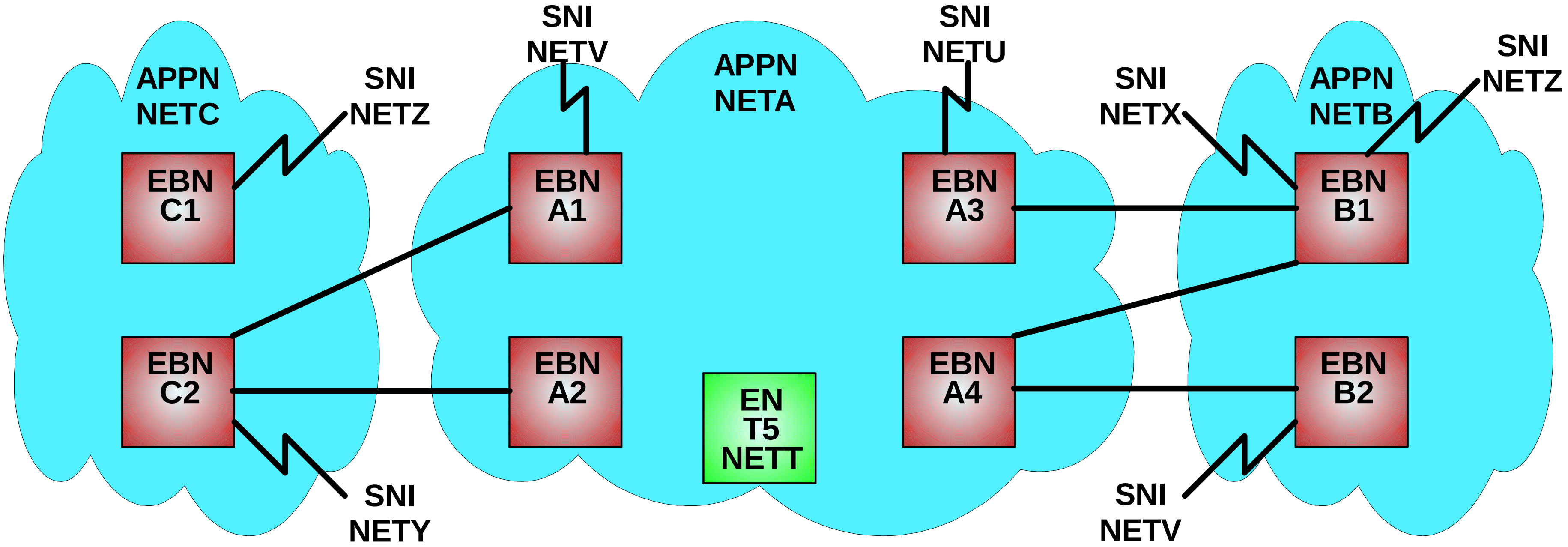
NETA.A4 ADJCLUST Tables (BNDYN=NONE)



```
ADJCLUST VBUILD    TYPE=ADJCLUST
NATIVE   NETWORK   NETID=(NETA,NETT,        *
                          NETU)
A4       NEXTCP    CPNAME=NETA.A4
*
THRUNETC NETWORK   NETID=(NETC,NETY)
A1       NEXTCP    CPNAME=NETA.A1
A2       NEXTCP    CPNAME=NETA.A2
*
THRUNETB NETWORK   NETID=(NETB,NETX)
B1       NEXTCP    CPNAME=NETB.B1
B2       NEXTCP    CPNAME=NETB.B2
A3       NEXTCP    CPNAME=NETA.A3
```

```
*
NETV     NETWORK   NETID=NETV
A4       NEXTCP    CPNAME=NETA.A4
B1       NEXTCP    CPNAME=NETB.B1
B2       NEXTCP    CPNAME=NETB.B2
A3       NEXTCP    CPNAME=NETA.A3
*
NETZ     NETWORK   NETID=NETZ
B1       NEXTCP    CPNAME=NETB.B1
B2       NEXTCP    CPNAME=NETB.B2
A3       NEXTCP    CPNAME=NETA.A3
A1       NEXTCP    CPNAME=NETA.A1
A2       NEXTCP    CPNAME=NETA.A2
```

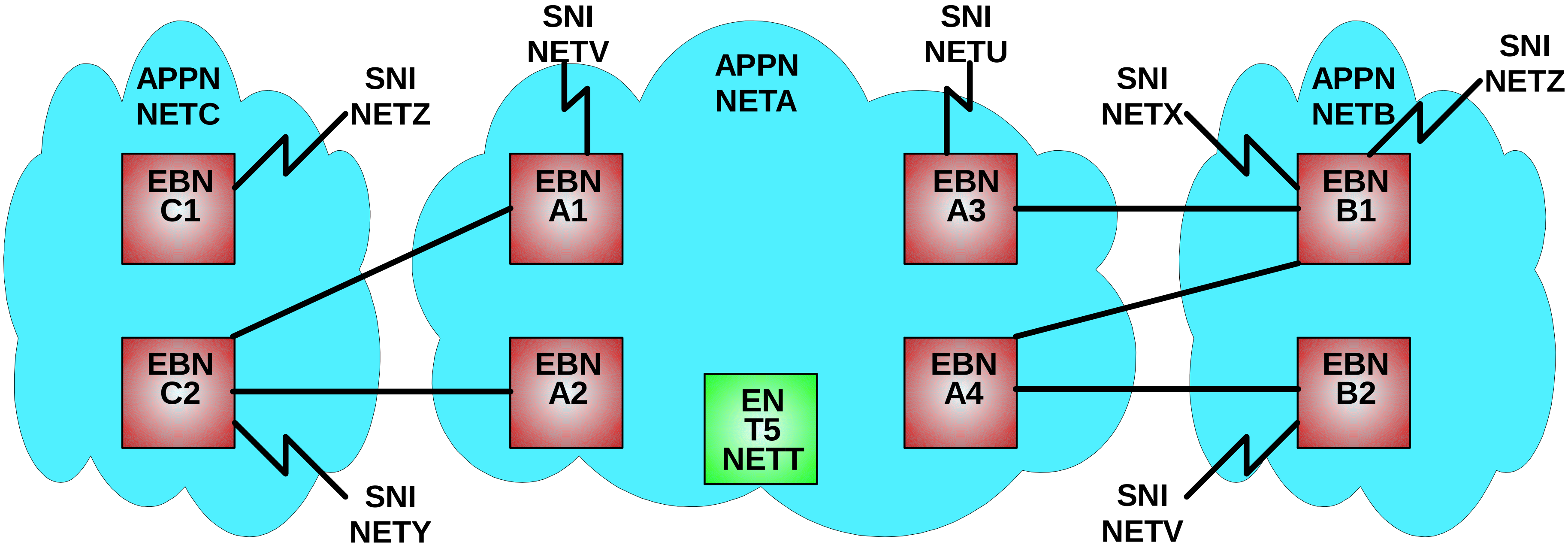
NETB.B1 ADJCLUST Tables (BNDYN=NONE) (a)



```
ADJCLUST VBUILD    TYPE=ADJCLUST
NATIVE   NETWORK   NETID=(NETB,NETX)
B1       NEXTCP    CPNAME=NETB.B1
*
THRUNETA NETWORK   NETID=(NETA,NETT,     *
                          NETU,NETC,     *
                          NETY)
A3       NEXTCP    CPNAME=NETA.A3
A4       NEXTCP    CPNAME=NETA.A4
B2       NEXTCP    CPNAME=NETB.B2
```

```
*
NETV     NETWORK   NETID=NETV
B1       NEXTCP    CPNAME=NETB.B1
A3       NEXTCP    CPNAME=NETA.A3
A4       NEXTCP    CPNAME=NETA.A4
B2       NEXTCP    CPNAME=NETB.B2
*
NETZ     NETWORK   NETID=NETZ
B1       NEXTCP    CPNAME=NETB.B1
A3       NEXTCP    CPNAME=NETA.A3
A4       NEXTCP    CPNAME=NETA.A4
B2       NEXTCP    CPNAME=NETB.B2
```

NETB.B1 ADJCLUST Tables (BNDYN=NONE) (b)



```
ADJCLUST VBUILD     TYPE=ADJCLUST
NATIVE   NETWORK    NETID=(NETB,NETX)
B1       NEXTCP     CPNAME=NETB.B1
*
THRUNETA NETWORK    NETID=(NETA,NETT,        *
                           NETU,NETC,        *
                           NETY)
A3       NEXTCP     CPNAME=NETA.A3
A4       NEXTCP     CPNAME=NETA.A4
B2       NEXTCP     CPNAME=NETB.B2
```

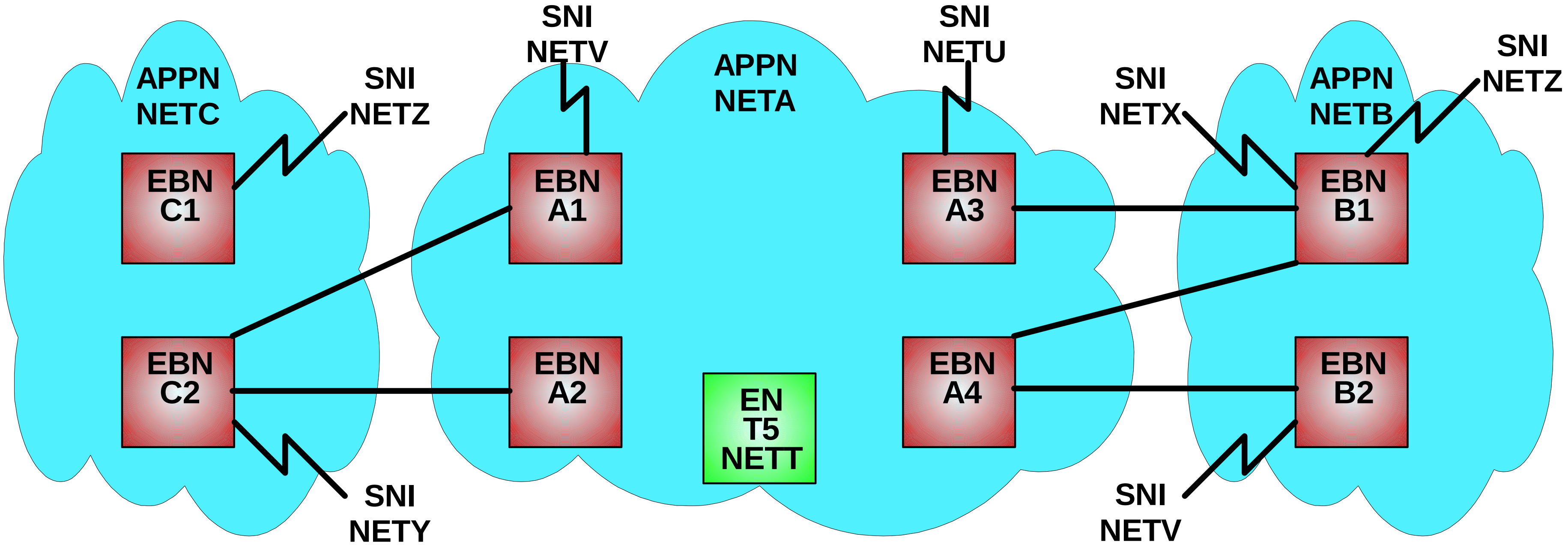
```
*
NETVZ    NETWORK    NETID=(NETV,NETZ)
B1       NEXTCP     CPNAME=NETB.B1
A3       NEXTCP     CPNAME=NETA.A3
A4       NEXTCP     CPNAME=NETA.A4
B2       NEXTCP     CPNAME=NETB.B2
```

Why might you want to split:

- NETA, NETT, NETU from NETC, NETY? ...or...
- NETV from NETZ?

SNVC=

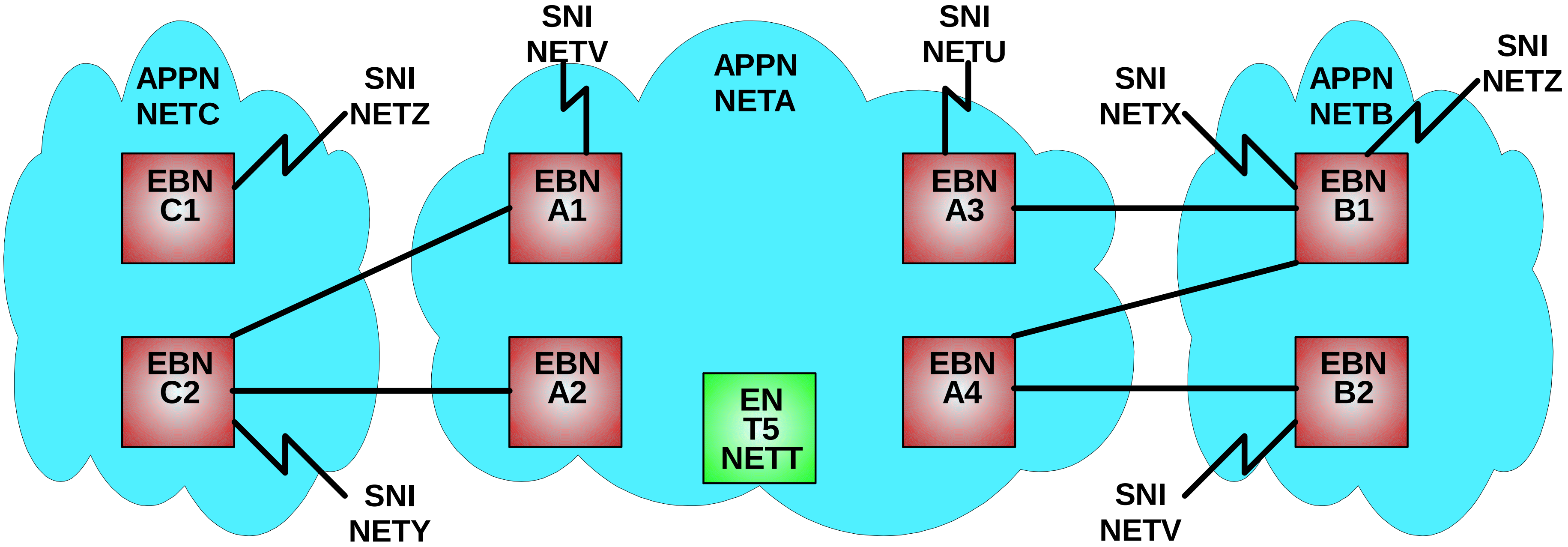
NETB.B1 ADJCLUST Tables (BNDYN=NONE) (c)



```
 ADJCLUST VBUILD     TYPE=ADJCLUST
 NATIVE   NETWORK    NETID=(NETB,NETX)
 B1       NEXTCP     CPNAME=NETB.B1
*
THRUNETA NETWORK    NETID=(NETA,NETT,       *
                           NETU,NETC,       *
                           NETY)
A3       NEXTCP     CPNAME=NETA.A3
A4       NEXTCP     CPNAME=NETA.A4
B2       NEXTCP     CPNAME=NETB.B2
```

```
*
NETV     NETWORK    NETID=(NETV)
B1       NEXTCP     CPNAME=NETB.B1
A3       NEXTCP     CPNAME=NETA.A3,SNVC=2
A4       NEXTCP     CPNAME=NETA.A4,SNVC=2
B2       NEXTCP     CPNAME=NETB.B2,SNVC=2
*
NETZ     NETWORK    NETID=(NETZ)
B1       NEXTCP     CPNAME=NETB.B1
A3       NEXTCP     CPNAME=NETA.A3,SNVC=3
A4       NEXTCP     CPNAME=NETA.A4.SNVC=3
B2       NEXTCP     CPNAME=NETB.B2,SNVC=3
```

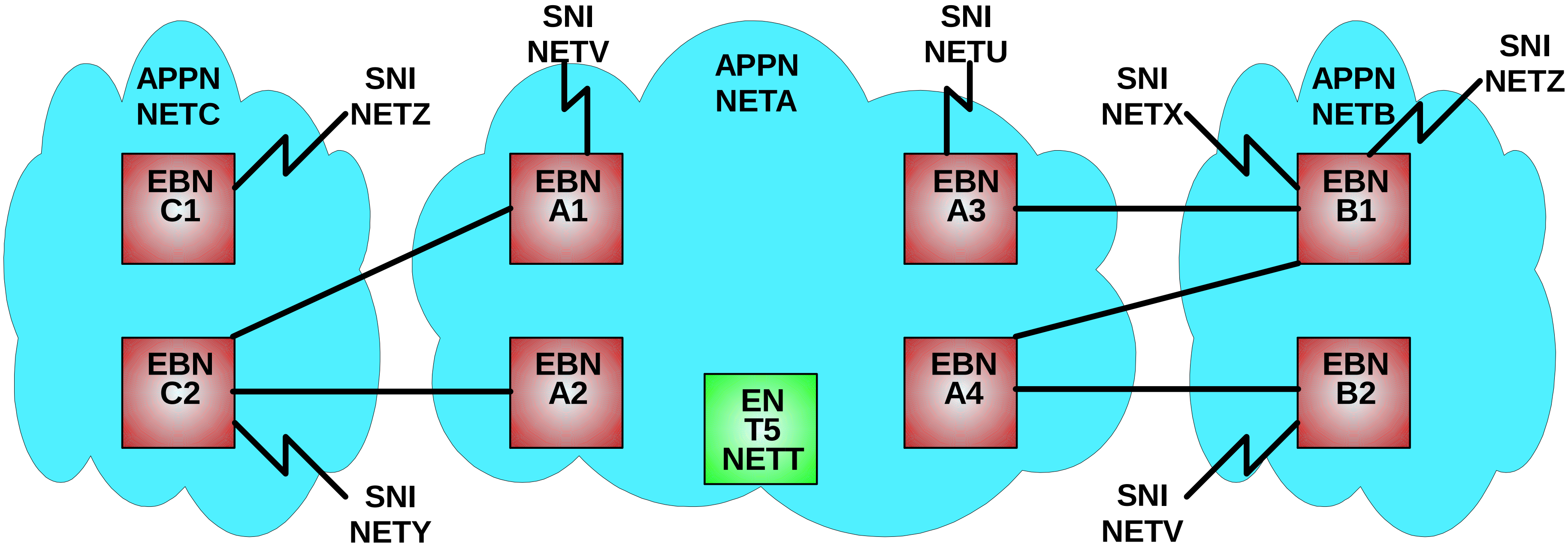
NETB.B2 ADJCLUST Tables (BNDYN=NONE)



```
ADJCLUST VBUILD     TYPE=ADJCLUST
NATIVE   NETWORK    NETID=(NETB,NETX)
B2       NEXTCP     CPNAME=NETB.B2
*
THRUNETA NETWORK    NETID=(NETA,NETT,      *
                           NETU,NETC,      *
                           NETY)
A4       NEXTCP     CPNAME=NETA.A4
B1       NEXTCP     CPNAME=NETB.B1
```

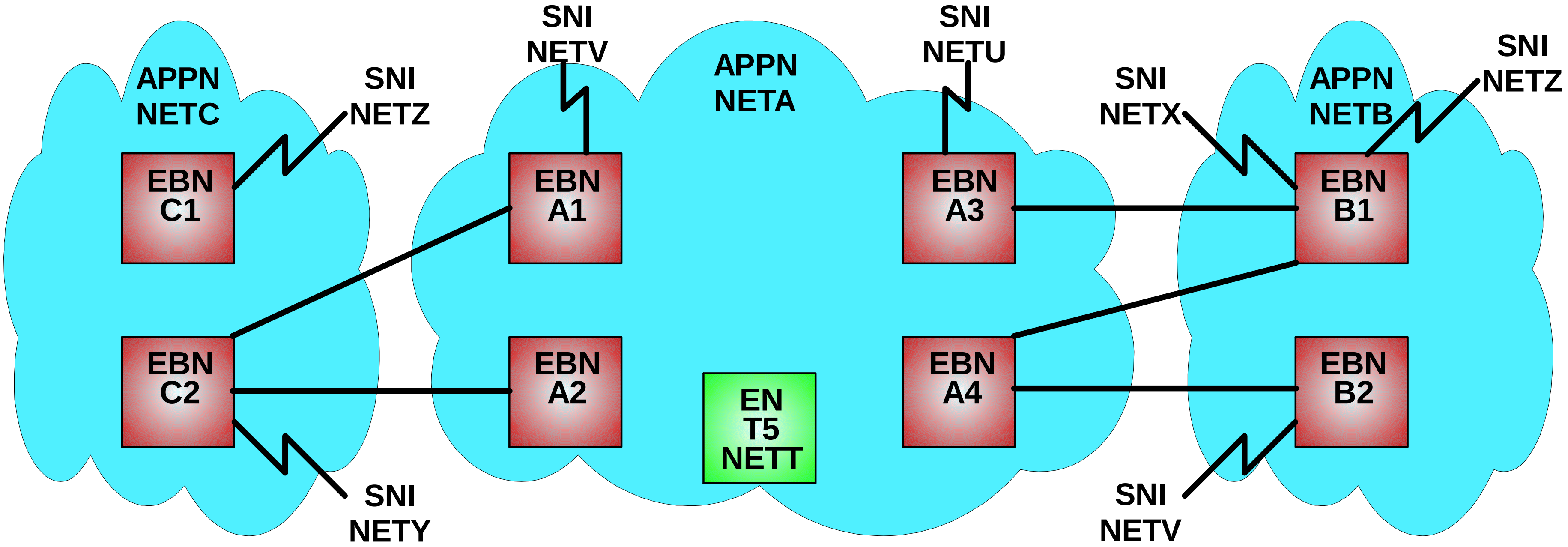
```
*
NETVZ    NETWORK    NETID=(NETV,NETZ)
B2       NEXTCP     CPNAME=NETB.B2
A4       NEXTCP     CPNAME=NETA.A4
B1       NEXTCP     CPNAME=NETB.B1
```

NETC.C1 ADJCLUST Tables (BNDYN=NONE)



```
ADJCLUST VBUILD    TYPE=ADJCLUST
NATIVE   NETWORK   NETID=(NETC,NETY)
C1       NEXTCP    CPNAME=NETC.C1
*
THRUNETA NETWORK   NETID=(NETA,NETT,     *
                          NETU,NETV,     *
                          NETB,NETX)
C2       NEXTCP    CPNAME=NETC.C2
*
NETZ     NETWORK   NETID=NETZ
C1       NEXTCP    CPNAME=NETC.C1
C2       NEXTCP    CPNAME=NETC.C2
```

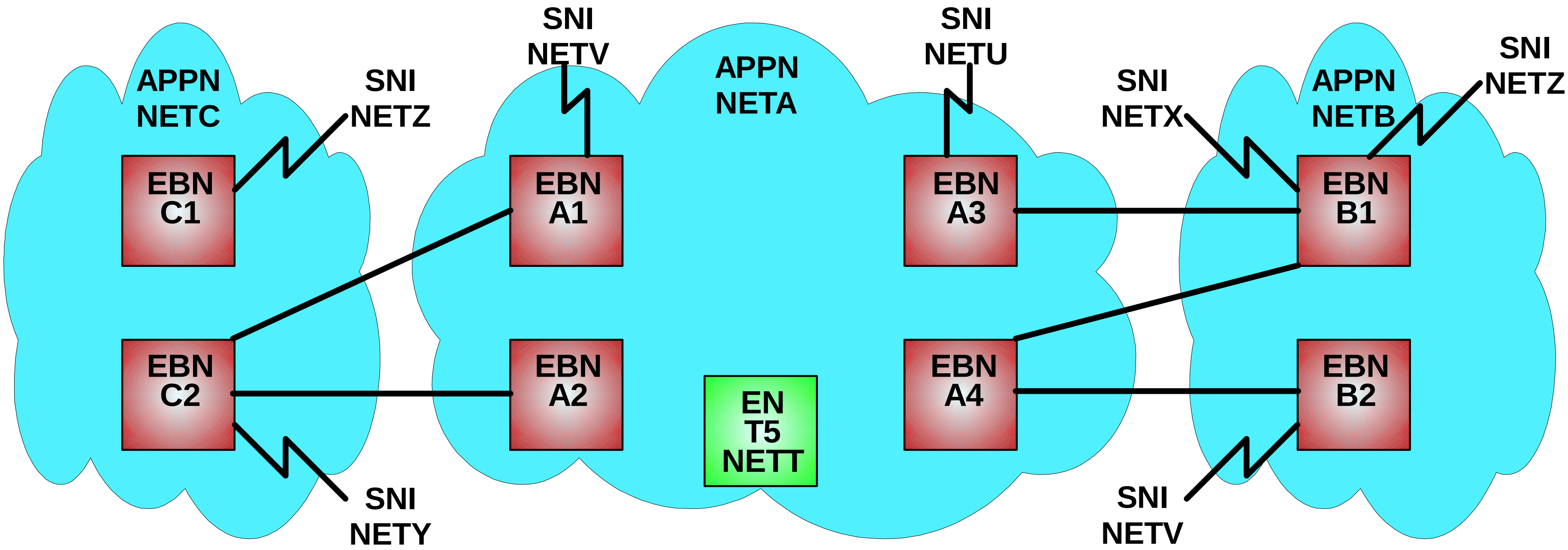
NETC.C2 ADJCLUST Tables (BNDYN=NONE)



```
ADJCLUST VBUILD    TYPE=ADJCLUST
NATIVE   NETWORK   NETID=(NETC,NETY)
C2       NEXTCP    CPNAME=NETC.C2
*
THRUNETA NETWORK   NETID=(NETA,NETT,        *
                          NETU,NETV,        *
                          NETB,NETX)
A1       NEXTCP    CPNAME=NETA.A1
A2       NEXTCP    CPNAME=NETA.A2
```

```
*
NETZ     NETWORK   NETID=NETZ
C2         NEXTCP    CPNAME=NETC.C2
A1         NEXTCP    CPNAME=NETA.A1
A2         NEXTCP    CPNAME=NETA.A2
```

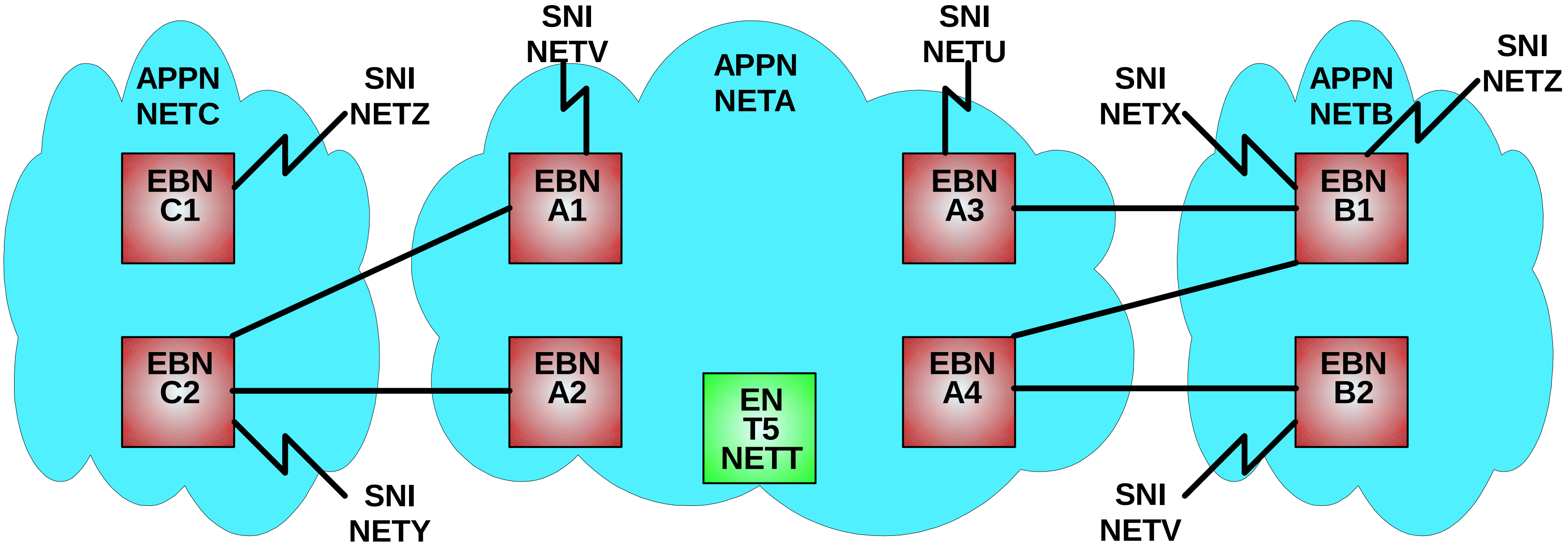
NETA.A1 Sample Default ADJCLUST Tables (a)



- All searches for non-native APPN resources use network-qualified real names
  - Restrict ALIAS searches to the native subnetwork only

```
ADJCLUST VBUILD     TYPE=ADJCLUST
DEFAULT  NETWORK    BNDYN=NONE
A1       NEXTCP     CPNAME=NETA.A1
```

NETA.A1 Sample Default ADJCLUST Tables (b)



- At least some searches for non-native resources will use ALIAS names
  - ALIAS searches must be allowed to search other subnetworks

```
ADJCLUST VBUILD     TYPE=ADJCLUST
DEFAULT  NETWORK    BNDYN=NONE
A1       NEXTCP     CPNAME=NETA.A1
C2       NEXTCP     CPNAME=NETC.C2
A2       NEXTCP     CPNAME=NETA.A2
A3       NEXTCP     CPNAME=NETA.A3
A4       NEXTCP     CPNAME=NETA.A4
```

or ...

```
ADJCLUST VBUILD     TYPE=ADJCLUST
DEFAULT  NETWORK    BNDYN=FULL
A1       NEXTCP     CPNAME=NETA.A1
```

Agenda

- Searching using ALIAS names
- Who is my network node server (NNS)?
- Introduction to Subarea and APPN searching
- Tuning Subarea Searching
  - Subarea search controls
  - ADJSSCP tables for sample subarea-to-APPN migration
- Tuning APPN EBN Searching
  - Introduction to extended border nodes (EBNs)
  - APPN search controls
  - Subnetwork routing lists (SRLs)
  - ADJCLUST tables for sample mixed APPN and subarea network
- Hints to aid in debugging search problems

Debug Aid: xSIRFMSG= Start Options

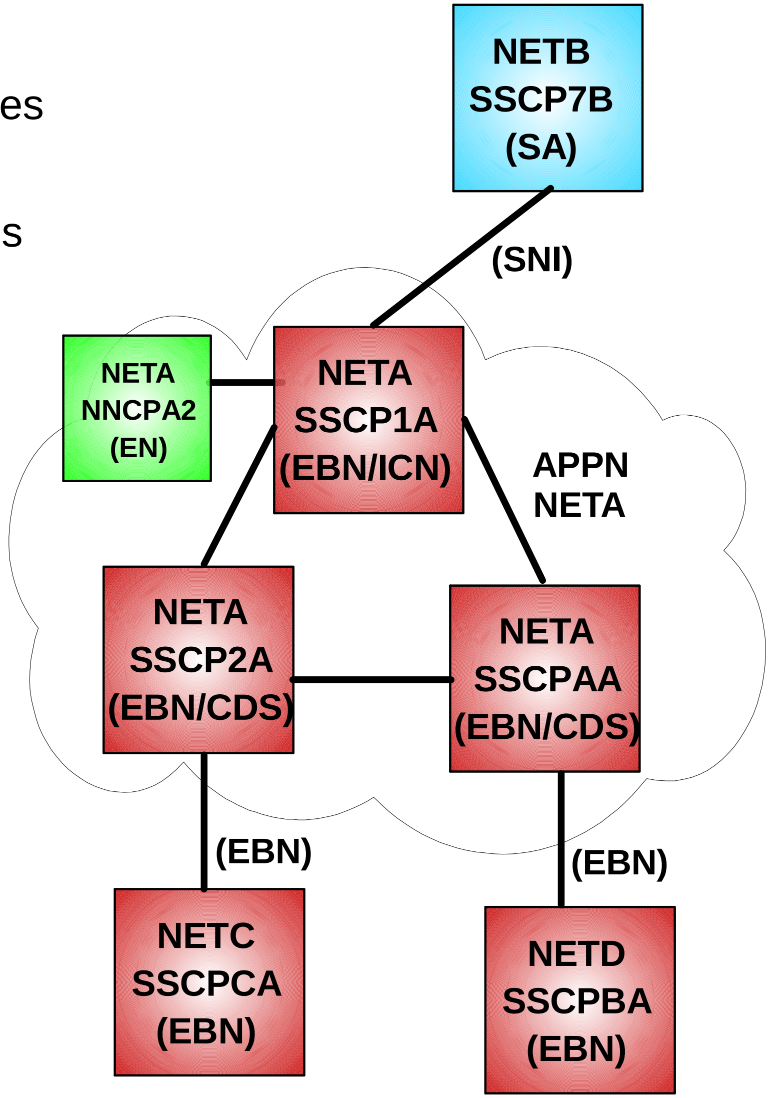
IST663I Message Group Options

- SIRFMSG=**ALLSSCP**, OLUSSCP or NONE
  - Controls output for session (INIT & CDINIT) searches
- DSIRFMSG=ALLSSCP, OLUSSCP or **NONE**
  - Controls output for non-session (DSRLST) searches
- LSIRFMSG=ALLNNS, OLUNNS or **NONE**
  - Controls output for APPN (Locate) searches

IST663I Message Subgroup Options

- FSIRFMSG=ALLSSCP, **OLUSSCP** or NONE
  - Controls output of search routing messages
  - Only displayed when "resource was not found"
- ESIRFMSG=**ALLSSCP**, OLUSSCP or NONE
  - Controls output of extended sense data
- ASIRFMSG=ALLSSCP, **OLUSSCP** or NONE
  - Controls output of Autologon messages

All xSIRFMSG Options Are MODIFYable!



Only SIRFMSG and FSIRFMSG Enabled

A search from SSCP1A for an unknown target LU results in the following messages:

On SSCP1A:

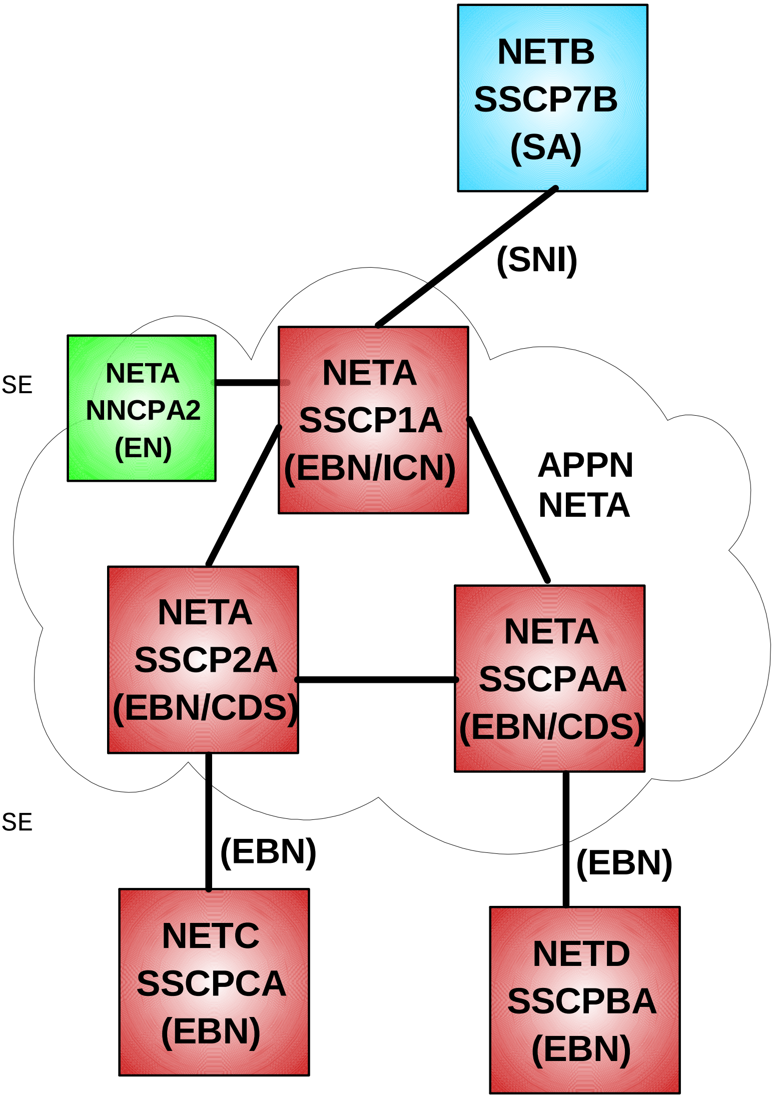
```
S IST663I INIT OTHER REQUEST FAILED, SENSE=087D0001
S IST664I REAL  OLU=NETA.APPL1            ALIAS DLU=NETA.YODA
S IST889I SID = EAABEEC3D39418A4
F IST1705I SORDER = APPN FROM START OPTION
F IST1705I SSCPORD = PRIORITY FROM START OPTION
F IST894I ADJSSCPS TRIED  FAILURE SENSE   ADJSSCPS TRIED  FAILURE SENSE
F IST895I    ISTAPNCP        08400007        SSCP7B          087D0001
S IST314I END
```

On SSCP7B:

```
S IST663I CDINIT REQUEST FROM SSCP1A FAILED, SENSE=087D0001
S IST664I REAL  OLU=NETA.APPL1            ALIAS DLU=NETA.YODA
S IST889I SID = EAABEEC3D39418A4
F IST1705I SORDER = APPN FROM START OPTION
F IST1705I SSCPORD = PRIORITY FROM START OPTION
F IST894I ADJSSCPS TRIED  FAILURE SENSE   ADJSSCPS TRIED  FAILURE SENSE
F IST895I    ISTAPNCP        08420001        SSCP1A          08260000
S IST314I END
```

Other hosts:

```
Nothing!  (Broadcast searches are handled as DSRLST searches!)
```



DSIRFMSG And LSIRFMSG Enabled Too

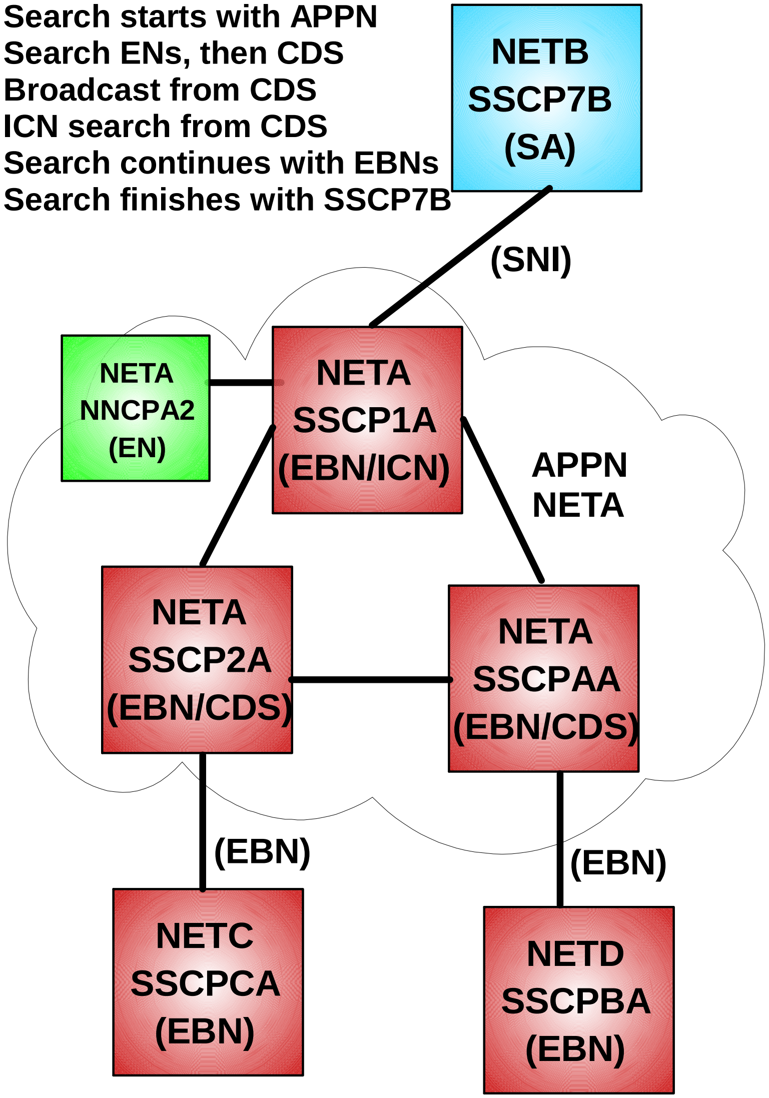
On SSCP1A:

```
L IST663I GDS LOCATE REQUEST FROM SSCP2A FAILED, SENSE=08400007     (3)
L IST664I REAL  OLU=NETA.SSCP1A         ALIAS DLU=NETA.YODA
L IST889I SID = EAABEEC3D39418A5
F IST1942I APPN LOCATE SEARCH STEPS ATTEMPTED
F IST1947I BROADCAST SEARCH TO SERVED END NODES
F IST1953I   NETA.NNCPA2        - SENSE 08400007 FROM NETA.NNCPA2
F IST1946I LOCAL SUBAREA SEARCH
F IST1953I   NETA.SSCP1A        - SENSE 08400007 FROM NETA.SSCP1A
L IST314I END

L IST663I GDS LOCATE REQUEST FROM SSCP2A FAILED, SENSE=08400007     (4)
L IST664I REAL  OLU=NETA.SSCP1A         ALIAS DLU=NETA.YODA
L IST889I SID = EAABEEC3D39418A5
F IST1942I APPN LOCATE SEARCH STEPS ATTEMPTED
F IST1946I LOCAL SUBAREA SEARCH
F IST1953I   NETA.SSCP1A        - SENSE 08400007 FROM NETA.SSCP1A
L IST314I END

L IST663I IPS SCR REQUEST FROM HOST CP FAILED, SENSE=08400007
L IST664I REAL  OLU=NETA.SSCP1A         ALIAS DLU=NETA.YODA
L IST889I SID = EAABEEC3D39418A5
F IST1942I APPN LOCATE SEARCH STEPS ATTEMPTED
F IST1947I BROADCAST SEARCH TO SERVED END NODES                     (2)
F IST1953I   NETA.NNCPA2        - SENSE 08400007 FROM NETA.NNCPA2
F IST1949I DIRECTED SEARCH TO A CENTRAL DIRECTORY SERVER            (2)
F IST1953I   NETA.SSCP2A        - SENSE 08400007 FROM NETA.SSCP1A
F IST1948I DIRECTED SEARCHES TO BORDER NODES                        (5)
F IST1953I   NETA.SSCPAA        - SENSE 08400007 FROM NETA.SSCPAA
F IST1953I   NETA.SSCP2A        - SENSE 08400007 FROM NETA.SSCP2A
F IST314I END
       :
       :
S IST663I INIT OTHER REQUEST FAILED, SENSE=087D0001
S IST664I REAL  OLU=NETA.APPL1          ALIAS DLU=NETA.YODA
S IST889I SID = EAABEEC3D39418A5
F IST1705I SORDER = APPN FROM START OPTION
F IST1705I SSCPORD = PRIORITY FROM START OPTION
F IST894I ADJSSCPS TRIED  FAILURE SENSE   ADJSSCPS TRIED  FAILURE SENSE
F IST895I   ISTAPNCP        08400007 (1)   SSCP7B          087D0001   (6)
S IST314I END
```

1. **Search starts with APPN**
2. **Search ENs, then CDS**
3. **Broadcast from CDS**
4. **ICN search from CDS**
5. **Search continues with EBNs**
6. **Search finishes with SSCP7B**



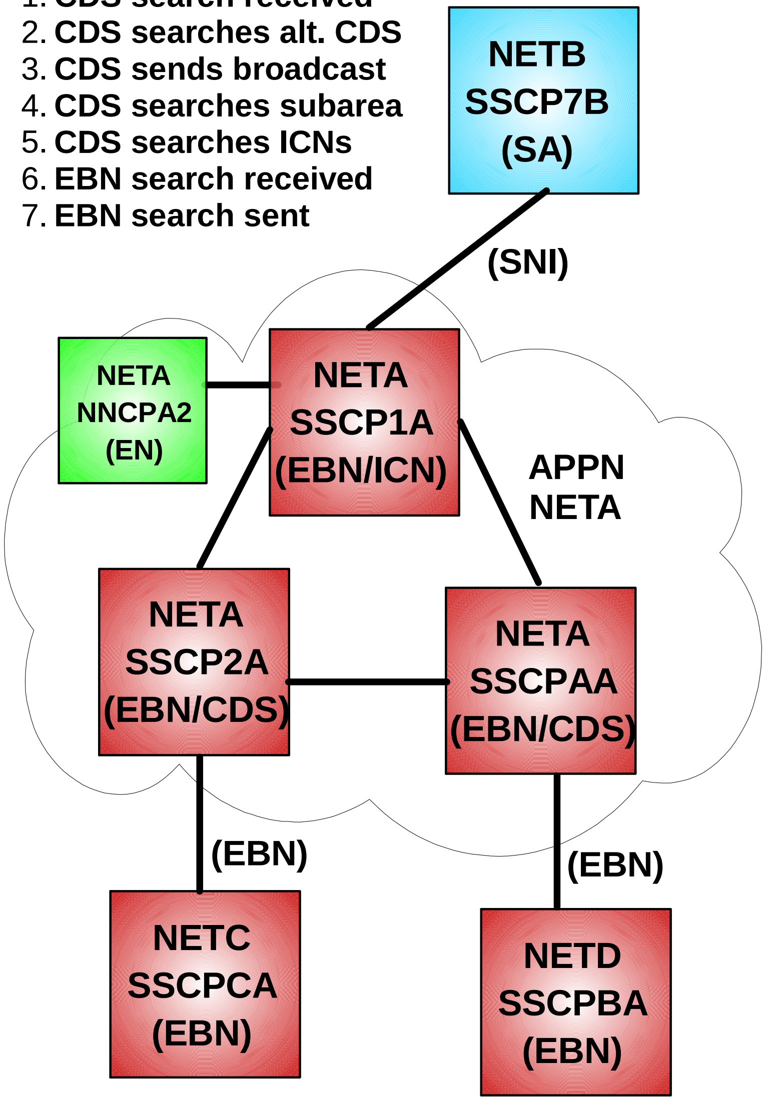
DSIRFMSG And LSIRFMSG Enabled ...

On SSCP2A:

```
D IST663I IPS SRQ REQUEST FROM ISTAPNCP FAILED, SENSE=087D0001        (4)
D IST664I REAL  OLU=NETA.SSCP1A         ALIAS DLU=NETA.YODA
D IST889I SID = EAABEEC3D39418A5
F IST1705I SORDER = APPN FROM START OPTION
F IST1705I SSCPORD = PRIORITY FROM START OPTION
F IST894I ADJSSCPS TRIED  FAILURE SENSE   ADJSSCPS TRIED  FAILURE SENSE
F IST895I    ISTAPNCP        08260000
D IST314I END

L IST663I GDS LOCATE REQUEST FROM SSCP1A FAILED, SENSE=08400007       (1)
L IST664I REAL  OLU=NETA.SSCP1A         ALIAS DLU=NETA.YODA
L IST889I SID = EAABEEC3D39418A5
F IST1942I APPN LOCATE SEARCH STEPS ATTEMPTED
F IST1950I DIRECTED SEARCHES TO ALTERNATE CENTRAL DIRECTORY SERVERS   (2)
F IST1953I   NETA.SSCPAA       - SENSE 08400007 FROM NETA.SSCPAA
F IST1951I BROADCAST SEARCH TO NETWORK NODES                          (3)
F IST1953I   NETA.SSCP1A       - SENSE 08400007 FROM NETA.SSCP1A
F IST1953I   NETA.SSCPAA       - SENSE 08400007 FROM NETA.SSCPAA
F IST1946I LOCAL SUBAREA SEARCH                                       (4)
F IST1953I   NETA.SSCP2A       - SENSE 08400007 FROM NETA.SSCP2A
F IST1952I DIRECTED SEARCHES TO INTERCHANGE NODES                     (5)
F IST1953I   NETA.SSCP1A       - SENSE 08400007 FROM NETA.SSCP1A
L IST314I END
                                                                      (6)
L IST663I GDS LOCATE REQUEST FROM SSCP1A FAILED, SENSE=08400007
L IST664I REAL  OLU=NETA.SSCP1A         ALIAS DLU=NETA.YODA
L IST889I SID = EAABEEC3D39418A5
F IST1942I APPN LOCATE SEARCH STEPS ATTEMPTED
F IST1946I LOCAL SUBAREA SEARCH
F IST1953I   NETA.SSCP2A       - SENSE 08400007 FROM NETA.SSCP2A
F IST1948I DIRECTED SEARCHES TO BORDER NODES                          (7)
F IST1953I   NETC.SSCPCA       - SENSE 08400007 FROM NETC.SSCPCA
L IST314I END
```

1. **CDS search received**
2. **CDS searches alt. CDS**
3. **CDS sends broadcast**
4. **CDS searches subarea**
5. **CDS searches ICNs**
6. **EBN search received**
7. **EBN search sent**



DSIRFMSG And LSIRFMSG Enabled ...

On SSCPAA:

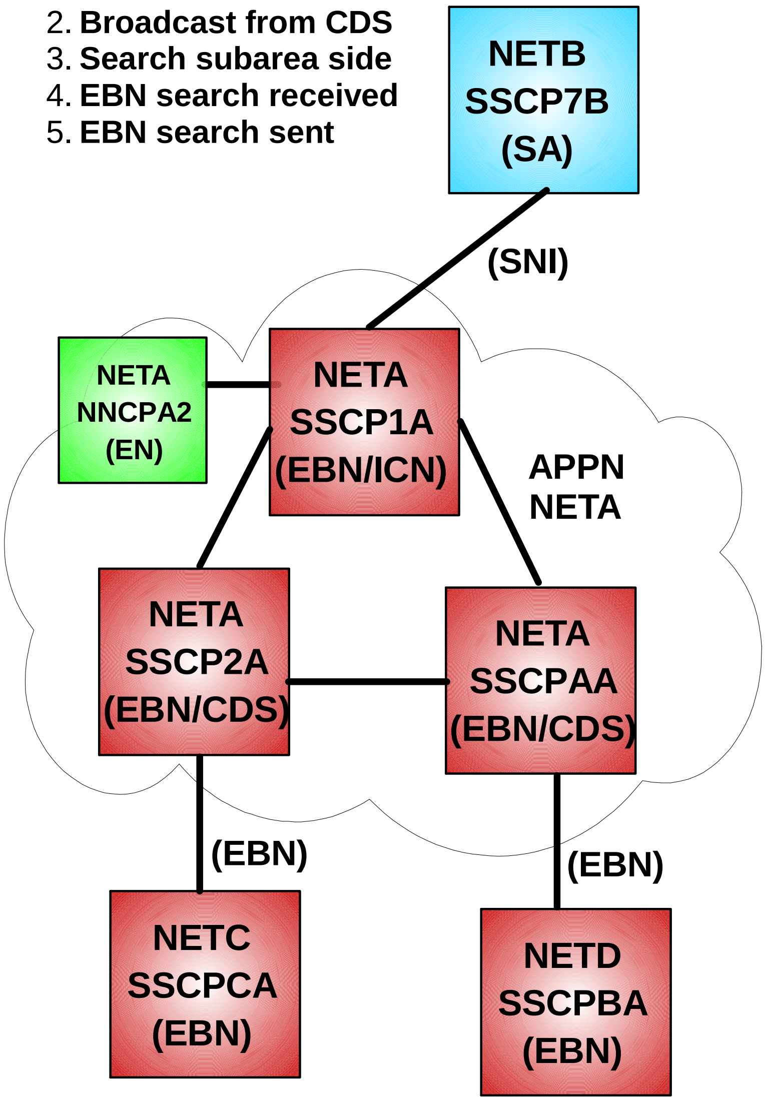
```
L IST663I GDS LOCATE REQUEST FROM SSCP2A FAILED, SENSE=08400007      (1)
L IST664I REAL  OLU=NETA.SSCP1A         ALIAS DLU=NETA.YODA
L IST889I SID = EAABEEC3D39418A5
L IST314I END

D IST663I IPS SRQ REQUEST FROM ISTAPNCP FAILED, SENSE=087F0004       (3)
D IST664I REAL  OLU=NETA.SSCP1A         ALIAS DLU=NETA.YODA
D IST889I SID = EAABEEC3D39418A5
D IST314I END

L IST663I GDS LOCATE REQUEST FROM SSCP2A FAILED, SENSE=08400007      (2)
L IST664I REAL  OLU=NETA.SSCP1A         ALIAS DLU=NETA.YODA
L IST889I SID = EAABEEC3D39418A5
F IST1942I APPN LOCATE SEARCH STEPS ATTEMPTED
F IST1946I LOCAL SUBAREA SEARCH                                      (3)
F IST1953I   NETA.SSCPAA        - SENSE 08400007 FROM NETA.SSCPAA
L IST314I END

L IST663I GDS LOCATE REQUEST FROM SSCP2A FAILED, SENSE=08400007      (4)
L IST664I REAL  OLU=NETA.SSCP1A         ALIAS DLU=NETA.YODA
L IST889I SID = EAABEEC3D39418A5
F IST1942I APPN LOCATE SEARCH STEPS ATTEMPTED
F IST1946I LOCAL SUBAREA SEARCH
F IST1953I   NETA.SSCPAA        - SENSE 08400007 FROM NETA.SSCPAA
F IST1948I DIRECTED SEARCHES TO BORDER NODES                         (5)
F IST1953I   NETD.SSCPBA        - SENSE 08400007 FROM NETD.SSCPBA
L IST314I END
```

1. **Alt. CDS search received**
2. **Broadcast from CDS**
3. **Search subarea side**
4. **EBN search received**
5. **EBN search sent**



DSIRFMSG And LSIRFMSG Enabled ...

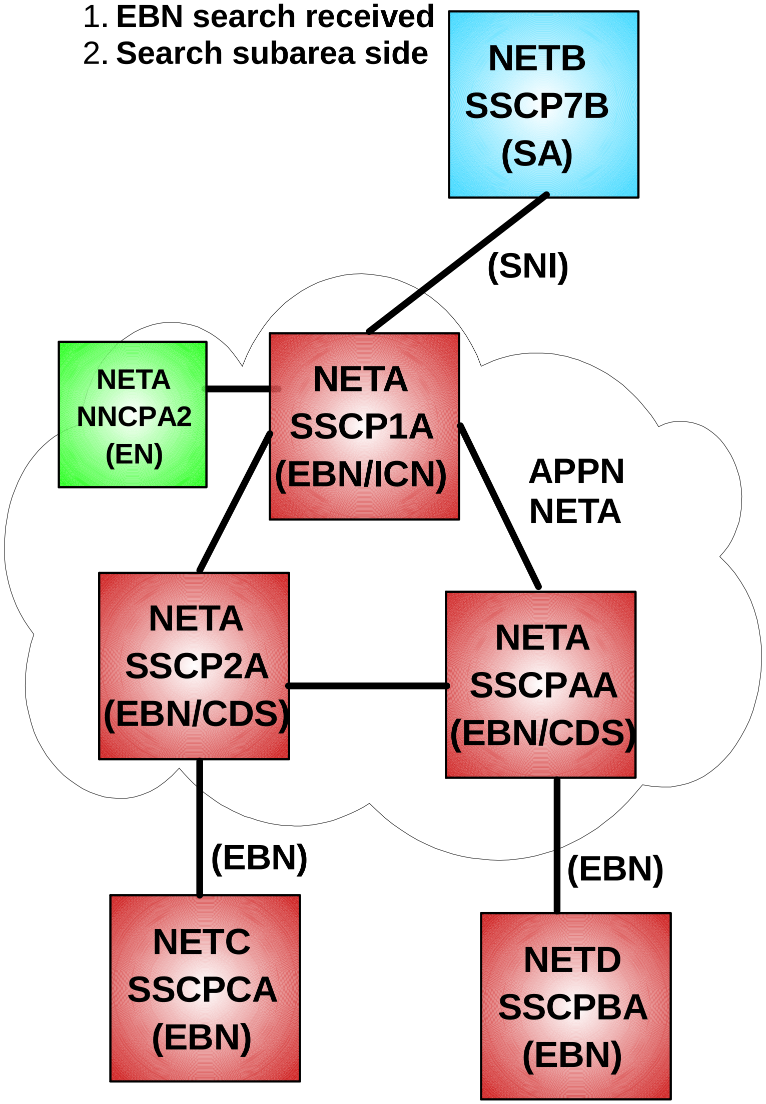
On SSCPBA (and SSCPCA):

```
D IST663I IPS SRQ REQUEST FROM ISTAPNCP FAILED, SENSE=087D0001        (2)
D IST664I REAL  OLU=NETA.SSCP1A         ALIAS DLU=NETA.YODA
D IST889I SID = EAABEEC3D39418A5
F IST1705I SORDER = APPN FROM START OPTION
F IST1705I SSCPORD = PRIORITY FROM START OPTION
F IST894I ADJSSCPS TRIED  FAILURE SENSE   ADJSSCPS TRIED  FAILURE SENSE
F IST895I    ISTAPNCP        08260000
D IST314I END

L IST663I GDS LOCATE REQUEST FROM SSCPAA FAILED, SENSE=08400007       (1)
L IST664I REAL  OLU=NETA.SSCP1A         ALIAS DLU=NETD.YODA
L IST889I SID = EAABEEC3D39418A5
F IST1942I APPN LOCATE SEARCH STEPS ATTEMPTED
F IST1946I LOCAL SUBAREA SEARCH                                        (2)
F IST1953I    NETD.SSCPBA       - SENSE 08400007 FROM NETD.SSCPBA
L IST314I END
```

On SSCP7B:

```
S IST663I CDINIT REQUEST FROM SSCP1A FAILED, SENSE=087D0001
S IST664I REAL  OLU=NETA.APPL1          ALIAS DLU=NETA.YODA
S IST889I SID = EAABEEC3D39418A5
F IST1705I SORDER = APPN FROM START OPTION
F IST1705I SSCPORD = PRIORITY FROM START OPTION
F IST894I ADJSSCPS TRIED  FAILURE SENSE   ADJSSCPS TRIED  FAILURE SENSE
F IST895I    ISTAPNCP        08420001        SSCP1A          08260000
S IST314I END
```



1. **EBN search received**
2. **Search subarea side**

Debug Aid: DISPLAY ADJSSCPS,CDRSC=

```
d net,adjsscps
 IST097I DISPLAY ACCEPTED
 IST350I DISPLAY TYPE = ADJACENT SSCP TABLE
 IST623I DYNAMIC ADJACENT SSCP TABLE
 IST1705I SORDER = APPN FROM START OPTION
 IST624I   SSCP2A
 IST1454I 1 RESOURCE(S) DISPLAYED
 IST314I END

d net,adjsscps,cdrsc=applaa1
 IST097I DISPLAY ACCEPTED
 IST350I DISPLAY TYPE = ADJACENT SSCP TABLE
 IST611I ADJACENT SSCP TABLE FOR APPLAA1 IN NETA
 IST1705I SORDER = APPN FROM START OPTION
 IST1220I  SSCPNAME NETID    CURRENT STATE  ROUTING STATUS
 IST624I   SSCPAA   NETA     NEVAC          ***NA**
 IST624I   ISTAPNCP NETA     ACTIV          ***NA**
 IST624I   SSCP2A   NETA     ACTIV          ***NA**
 IST1454I 3 RESOURCE(S) DISPLAYED
 IST314I END
```

- **DISPLAY ADJSSCPS,CDRSC=resource**
  - Displays ADJSSCP Table (In Order) For **This CDRSC!!**
    - Takes SORDER And SSCPORD Values Into Account
  - Use To Determine Which CDRMs (And/Or APPN) Will Be Searched
    - For The Next Search (May Not Be The Same As The Previous Search)
    - Use The Same Command At Those Other CDRMs (To Follow The Trail)!
  - What If CDRSC Does Not Exist?
    - Use MODIFY ALSLIST,...,ACTION=CREATE

Debug Aid: DISPLAY SRCHINFO

```
D NET,SRCHINFO,LIST=SUMMARY
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = SRCHINFO
IST1520I SUBAREA SEARCH INFORMATION:
IST1521I   TOSSCP   NAME   CDINIT    DSRLST    IOCD   INITOT     TOTAL
IST1522I   SSCP1A               0         1       0        2         3
IST1525I   TOTAL NUMBER OF OUTSTANDING SEARCHES =          3
IST1454I 1 SSCP NAME(S) DISPLAYED
IST924I --------------------------------------------------------------
IST1526I APPN SEARCH INFORMATION:
IST1527I   TOCP   NAME       TYPE  STATUS   BROADCAST DIRECTED   TOTAL
IST1528I   NETA.SSCPAA        NN    OPEN            2        1       3
IST1525I   TOTAL NUMBER OF OUTSTANDING SEARCHES =          3
IST1454I 1 CP NAME(S) DISPLAYED
IST314I END

D NET,SRCHINFO,LIST=ALL
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = SRCHINFO
IST1520I SUBAREA SEARCH INFORMATION:
IST1523I       OLU                DLU                SID       RU
IST1524I   NETA.APPL1        NETA.SSCP1A       ****NA****      DSRLST
IST1524I   NETA.APPL2        NETA.SSCP1A       ****NA****      INITOT
IST1524I   NETA.SSCP1A       NETA.SSCP1A       ****NA****      INITOT
IST1454I 3 PAIR(S) DISPLAYED
IST924I --------------------------------------------------------------
IST1526I APPN SEARCH INFORMATION:
IST1529I       OLU                DLU                SID       LOCATE
IST1530I   NETA.SSCP1A       NETA.SSCP2A       EAABEEC3C6093893     1
IST1530I   NETB.SSCP7B       NETA.NETAPPL2     C2BB19BCF437741D     1
IST1530I   NETB.SSCP7B       NETA.NETAPPL1     C2BB19BCF437741C     1
IST1525I   TOTAL NUMBER OF OUTSTANDING SEARCHES =          3
IST1454I 3 PAIR(S) DISPLAYED
IST314I END
```

- **DISPLAY SRCHINFO** Command
  - Displays All Outstanding APPN And Subarea Searches
    - CDINITs, DSRLSTs, IOCDs, APPN Locates
  - Not All Can Be Terminated By Operator Command
    - For Example: DSRLSTs, IOCDs, Some APPN Locates

Debug Aid: DISPLAY SRCHINFO,SID=

```
D NET,SRCHINFO,SID=EAABEEC3C6093891
IST097I DISPLAY ACCEPTED
IST350I DISPLAY TYPE = SRCHINFO
IST1520I SUBAREA SEARCH INFORMATION:
IST1531I   SID = EAABEEC3C6093891         CP(OLU) = SSCP1A
IST1532I   OLU = NETA.APPL1               DLU = NETA.NETAPPL1
IST1540I   SEARCH STATUS = PDSRLST        SSCP(OLU) = SSCP1A
IST1539I   PCID MODIFIER = 00000000000000000000
IST1534I   SSCP/CP IN OLU DIRECTION = NETA.SSCP1A
IST1533I   SEARCH CONCENTRATED = NO       RDS = NO
IST1705I SORDER = APPN FROM START OPTION
IST894I ADJSSCPS TRIED  FAILURE SENSE   ADJSSCPS TRIED  FAILURE SENSE
IST895I    SSCP2A          08420000
IST1454I 1 ADJSSCP(S) DISPLAYED
IST1537I   AWAITING REPLY FROM THE FOLLOWING NODE(S):
IST1538I      NETA.ISTAPNCP
IST924I -------------------------------------------------------------
IST1526I APPN SEARCH INFORMATION:
IST1531I   SID = EAABEEC3C6093891         CP(OLU) = NETA.SSCP1A
IST1532I   OLU = NETA.APPL1               DLU = NETA.NETAPPL1
IST1539I   PCID MODIFIER = 10000000000000000000
IST1545I   NODE ROLE VECTOR = X'A000'
IST1541I   LOCATES PENDING =          0   CURRENT TASK = X'0B'
IST1533I   SEARCH CONCENTRATED = YES      RDS = NO
IST1534I   SSCP/CP IN OLU DIRECTION = NETA.SSCP1A
IST1535I   REPLY RETURNED TO ORIGINATING CP = NO
IST1536I   CONCENTRATED BEHIND C2BB19BCF437741C 22100000000000000000
IST924I -------------------------------------------------------------
IST1531I   SID = C2BB19BCF437741C         CP(OLU) = NETB.SSCP7B
IST1532I   OLU = NETB.SSCP7B              DLU = NETA.NETAPPL1
IST1539I   PCID MODIFIER = 22100000000000000000
IST1545I   NODE ROLE VECTOR = X'2000'
IST1541I   LOCATES PENDING =          1   CURRENT TASK = X'17'
IST1533I   SEARCH CONCENTRATED = NO       RDS = YES
IST1548I   BROADCAST = YES                DIRECTED = NO
IST1534I   SSCP/CP IN OLU DIRECTION = NETB.SSCP7B
IST1535I   REPLY RETURNED TO ORIGINATING CP = NO
IST1537I   AWAITING REPLY FROM THE FOLLOWING NODE(S):
IST1538I      NETA.SSCPAA
IST1543I   REQUESTS CONCENTRATED BEHIND THIS SEARCH =     1
IST314I END
```

For More Information...

URL	Content
http://www.ibm.com/systems/z/	IBM Mainframe
http://www.ibm.com/systems/z/hardware/networking/index.html	IBM Mainframe Networking
http://www.ibm.com/software/network/commserver/	Communications Server product overview
http://www.ibm.com/software/network/commserver/zos/	z/OS Communications Server overview
http://www.ibm.com/software/network/commserver/z_lin/	Communications Server for Linux on system z
http://www.ibm.com/software/network/ccl/	Communication Controller for Linux on system z
http://www.ibm.com/software/network/commserver/library/	Communications Server products - white papers, product documentation, etc.
http://www.ibm.com/systems/z/os/zos/bkserv/	z/OS Internet library - PDF versions of z/OS manuals (including z/OS CS)
http://www.redbooks.ibm.com	ITSO Redbooks
http://www.ibm.com/software/network/commserver/support	Communications Server technical Support
http://www.ibm.com/support/techdocs/atsmastr.nsf/Web/TechDocs	Technical support documentation from ATS (techdocs, flashes, presentations, white papers, etc.)
http://www.rfc-editor.org/rfcsearch.html	Request For Comments (RFC)
http://publib.boulder.ibm.com/infocenter/ieduasst/stgv1r0/index.jsp	IBM education assistant