## APPENDIX A. BIBLIOGRAPHY

An overview of the facilities available in the IBM X.25 NPSI PP is available in;

- X.25 NPSI General Information (GC30-3080)

A detailed technical discussion of the X.25 interface can be found in;

- The X.25 Interface for attaching IBM SNA Nodes to Packet-Switched Data Networks. General Information Manual. (GA27-3345)

The generation macro operands for the NPSI are documented in;

- X.25 NCP Packet Switching Interface - Installation and Operation. SC30-3163. (Releases 2 and 3.)

A detailed technical discussion on the installation experience of the NPSI in an OS/VS environment;

- X.25 NPSI Release 2 and 3 Guide. (GG24-1567)

A detailed technical discussion on SNA-to-SNA connection over an X.25 network including implementation, performance tuning and problem determination considerations;

- X.25 SNA Guide. (GG24-1568)



SHARE SESSION REPORT

| 61 | C155 | SNA Session Establishment White Paper | | 50 |
|---|---|---|---|---|
| SHARE NO. | SESSION NO. | SESSION TITLE | | ATTENDANCE |
| TPAM | | | Jim Cobban | DCL |
| PROJECT | | | SESSION CHAIRMAN | INST. CODE |

Datacrown Inc. 650 McNicoll Ave. Willodale, ON, Canada (416) 499-1012

SESSION CHAIRMAN'S COMPANY, ADDRESS, AND PHONE NUMBER

The final report of the SNA Session Establishment White Paper Committee was presented. This paper addresses deficiencies in the currently available products for handling logons to applications in a VTAM, TCAM or other SNA Network. There was a breif foil presentation explaining the nature of the problems, reviewing existing solutions, and giving an overview of the White Paper. The remainder of the meeting was a question and answer session on the paper.

SHARE TPAM PROJECT

SNA Session Establishment Control

Draft Report

February 14, 1983





Objective:

Identify a set of user requirements for functional enhancements in the area of facilitating and controlling the establishment of sessions between logical units in an SNA network.

Scope:

This paper does not consider the question of termination of sessions. However, it is felt that this should be more consistent between various applications.

Some interest was expressed in providing an ability on the initial session establishment to specify an application to be invoked at a more detail level. An example of this would be allowing the end-user to specify "ADRS" and get straight in, as opposed to asking to TSO or CMS, then ask that level of system for APL, loading the appropriate APL library and then invoking ADRS. This sounds like a good idea but is beyond the functionality that can be achieved without major rewrites to every interactive subsystem. This topic is therefore recommended to the Interactive Systems Task Force for study.

## 2. Summary

IBM should supply a session control interface product or products which will supplement Unformatted System Services and Formatted System Services of VTAM and TCAM, and some of the Message Control Program functions of TCAM. This interface will monitor all session requests in a SNA network. The session control interface might be implemented either as a direct extension to USS, in the VTAM/TCAM SSCP, or as a separate application, along the lines of the original Network Solicitor.

The purpose of this product is to enhance the appearance of a single network image to the end-user during session initiation. This is very important in a complex network connecting multiple geographically dispersed nodes. The function includes providing as similar as possible a session establishment interface to all terminal users as possible. The user should not have to learn a device-dependent session establishment process. As far as possible there should be a single step session establishment process in which all of the data necessary to place the user in session with the application is collected in one up front step by the session control product, and then passed with the session establishment request to the application.



In addition to whatever facilities may be specified in table form, there must be the ability to invoke user exit routines, both directly out of the session control product, and out of RACF. It is desireable that these exits be programmeable in a higher level language such as COBOL, FORTRAN, or PL/I. The exit routines should be invoked asynchronously to the access method.

A compatible session control interface should be available in all teleprocessing access methods which support SNA and in all programmeable cluster nodes. It must run under all IBM operating systems which support SNA access methods. This includes the operating systems running on distributed data processing systems such as the 8100. Where multiple processors are involved, the session control interface product on each processor should cooperate with its counterparts on the other processors in routing a session.

It must support all terminals which are supported by the above TP access methods, including SNA, BSC, and start/stop protocols. All terminals will be supported in as device independent a manner as possible without sacrificing the ability to utilize the power and flexibility of some terminals such as 3270s. Where a terminal product provides a method of constructing a Formatted System Services initiate from data provided by the end-user, the appearance of the



interface provided to the end-user should be compatible with that provided by the session control interface product.

This implies that the session control product may be a series of products with pieces running in each of the programmable nodes in the network. These pieces should cooperate to achieve the appearance of a single, attachment independent, session establishment interface.

Operation of the session control interface should be possible through the same command interfaces as for the access method.

a) End User Interface

i) Consistency

The session establishment procedure should be as consistent as possible across the range of supported terminals. There should not be any difference in the procedure for establishing a session between any supported display terminal and an application.

On display terminals the end user should receive a full screen display prompting simultaneously for all permissable input fields, and displaying all desired output information. If all of the necessary data is entered on this full screen, and is valid for the application, the user should not be reprompted by the application to reenter the data.

Prompting for non-display type devices should allow for entering the multiple possible input fields without requiring prompting. If insufficient data is supplied, however, the user should be prompted. Once again, as far as possible the output presented to the end user, and the input actions required must be as consistent as possible across the whole range of supported terminals.



Justification

There is too much variation in methods of logging on for various types of terminals. Even within a single type of terminal, for example a 3276 or 3767, the actions required of the end user to log on depend upon the setting of a switch on the terminal. The end user should not have to learn a range of different session establishment procedures.

At present the required session establishment process depends upon the method of attachment of the terminal. If a 3277/8 is attached to a SNA 3274/6 one method is used, if attached to a BSC/LOCAL 3274/6 another, if attached to an 8100 another, if attached to a 3600 yet another! Unless an external label has been attached to the terminal to let the user know how it is hooked up there is no way to know how to log on.

One of the objections raised to the use of SNA is that, in return for being given the flexibility to choose a target application, the user is forced to respond to an additional prompt in all circumstances, even on terminals which are only very infrequently used on anything other than the production application. If the application prompting could be consolidated into the session control interface prompt the annoyance level is decreased.



Suggested Implementation

The data stream used for establishing sessions for all display terminals should be LU2, or a mapping of LU2 into the device dependent character streams of the non-3270 display terminals. The data stream mapping might be done in a front-end processor. The data stream used for establishing sessions for non-display terminals should be LU1 or a mapping of LU1 into the device dependent data streams as above.

ii) Presentation Services

The design of a display for session establishment prompting should be as flexible as possible. The screen layouts should be designed interactively. At the discretion of the installation it should be possible to have multiple screen output layouts or line by line prompting formats.

As a minimum, in addition to being able to specify any desired constant data, the user should be able to select from the following variable output fields:

1) logical unit name of the terminal
2) current date
3) current time, to be updated when the screen is refreshed
4) an installation specified identification of the control point in the network for this terminal. This might be a telephone number for the network Hotline, or some node identification. In a network which has multiple control points handling problem calls, which occasionally change domain boundaries, it might be the name of the control point, or the name of the city containing the control point.
5) a set of application names or identifiers. For example; a session establishment screen may be set up which permits logging on through the use of a selector pen, or program function keys.
6) the current status associated with each of the above applications. These statuses should only be refreshed upon request from the user, or upon re-display of the menu after the user logs off another application.
7) date of most recent broadcast news item (see below).
8) diagnostic messages
9) network news area, multiple lines (see below).

The user should be able to specify names for input fields on the display, and the information on what input fields have been entered, and their values, should be supplied to a user exit. The one presumeably essential field would be the application name. The action of the user exit is described in the following section.

It must be possible to specify the action to be taken when various attention keys are pressed on terminals which support them, and to specify the action to be taken, or the interpretation to be made, of the use of a light pen.





Justification

The intention of this facility is to minimize the amount of central network administration, while enhancing end-user control. The existing unformatted system services is completely inadequate, in as much as it is limited to 256 byte messages and is forced to use SLU1 data streams even to some 3270s.

Suggested Implementation

Presentation services map design should use some preexisting presentation services design product, or a successor to such a product, rather than having a design function local to session control.

The system should support multiple screen layouts, with the end user being able to select which of the layouts best suits his/her requirements, by command entered at the terminal. If authorized to do so the end user should be able to invoke the screen design process from the Session Controller main menu, create a new layout, and assign it a unique name, without involving central site system programmers. The session control interface should reject a request for a named screen layout which is not known to the system, or for a named layout which cannot be displayed at the particular terminal, for example because it requires a larger presentation space, or for a named layout for which authorization was denied by an installation exit.

Once designed, the screen layouts should be presented in as compatible a manner as possible on all display terminals, regardless of size, shape, or functional capabilities. For example it may be desireable to be able to specify that the specified session establishment display will always be centered in the presentation space, rather than appearing in the upper left corner. It should be possible to specify extended attributes, such as colour, or highlighting, where the terminals are capable of implementing then. The presentation services interface should determine the capabilities of the particular terminal, and attempt to map the screen layout in as compatible a manner as possible.

The interactive map development capability of DMS or DPPX/DPS is perhaps the most user-friendly implementation of such a function. An alternative might be the menu management of SPF, although this would require enhancements in support of extended attributes.

iii) Application Specification

The installation should be provided with a choice of a powerful set of translation tables and exit routines for interpreting application names. The exit routines should be supplied with all of the various input fields, as defined by the presentation services map definition, in order to make the evaluation of which application the end user wishes to use. The routine may return specifying:

1) An application identifier which the access method can resolve directly, or ...
2) An indication that certain specified additional input fields must be supplied which were either ommitted by the user on the first try, or, for non-display terminals, have not been prompted for yet. The end-user will then be prompted for those specific fields. When the values have been obtained, the exit will be redriven. or...
3) An indication that the request has been rejected. In this case a message number will also be returned. The specified message will be returned to the user to explain the reason for the failure. It is not adequate to return "REJECTED BY INSTALLATION EXIT" to the terminal user as is done by RACF.



Justification

The existing functions under VTAM, Interpret and Unformatted System Services are insufficiently flexible and restrict the functions which can be performed by the exit routines. They are also an integrity exposure, running as they do in the access method address space. By moving the functions out of the access method it may become possible to distribute some of this function to less sophisticated users, or to management and so reduce network administration overhead, and the use of scarce and expensive system programmers. The application interpretation function is currently inconsistent between VTAM and TCAM. In reporting problems with a session establishment request to the terminal user, insufficient information is provided to allow the user to take corrective action. Typically all that is received is "SESSION NOT BOUND".



Suggested Implementation

The exits should be invoked asynchronously to the rest of the access method processing. The exits should run either in another address space, or partition, or if run in the access method address space, in a different protect key from the access method. The exits must be able to request operating system services without interfering with the operation of the access method. It is highly desireable that it be possible to write the exits in a higher level language.

iv) Userids and Passwords

The concept of a user identification, with associated password, is common to almost all functions which have a security requirement, such as TSO, CMS, IMS, CICS, NCCF, Info/Management and a host of non-IBM products. There has been little consistency even in the format of these userids. The userids should be as consistent and universal as possible for all applications.

The SNA session control interface should collect accounting information on users logging on through the network and should verify users via a password. The userid can then be used for billing back network usage. See the SNA Network Accounting and Performance White Paper. The userid and password should be passed through to the application where they should be used, rather than reprompting for an application specific userid and password. Userid and password verification should be external to the session control function. The userid verification might fail for several reasons, which must be spelled out to the end user. For example:

1) invalid password
2) password has expired
3) userid not authorized to access this application
4) the maximum number of users of the specified application permitted for the group or supergroup of userids to which this user belongs has been exceeded.
5) userid already logged on to this or another application.
6) (on reconnect) userid not found

Existing applications should reject attempts by the user to bypass this verification by logging on without first logging off and passing back through the session control interface.

The session control interface should keep track of which terminals particular users are using in the network, and to which applications they are connected. Optionally a particular userid should not be permitted to be on more than one application at a time. For some applications (for example TSO) it is possible to disconnect the terminal from the application while leaving a transaction running, and then reconnect at some later time. The reconnect function must be able to locate the application the userid is on wherever it is either within a computing complex or, optionally, in the whole network, and reestablish the connection without the end-user having to remember where the original session was.





Justification

In a complex network involving multiple hosts, or multiple sites, there is a requirement to account for network traffic, since access method cycles, front-end processors and trunks are being shared between various users.

The interpretation of the entered application name may depend upon the identity of the user making the request. For example routing a TSO user to a system which has access to his disk data sets.

There is excessive duplication of administrative effort having to maintain independent definitions of users on multiple products. Also the use of a single global userid on all data communications products would facilitate the design of electronic mail systems.

Suggested Implementation

The userid and password verification process should be external to the session control interface. An asynchronous user exit should be provided which could be used to translate common network userids to application dependent userids and passwords. This exit should be able to request normal operating system services such as I/O. In addition the session controller could optionally call RACF. RACF may already be used to maintain the userids for TSO, CICS, and IMS.

v) Error Messages

Error message texts must be easily customizeable. Additional messages may be added in reserved message number ranges. The message number should be part of the customizeable message text so the administrator may alter or remove it altogether. The administrator may change the text of a message in the IBM portion of the list of messages, and may delete or add symbolic substitutions within a message text.

Error messages issued to the end user should be as detailed and specific as possible. The user should be able to invoke a "help" function to obtain additional information about the message, such as interpretation of return codes, or error flags contained in the message.

Justification

The messages issued by USS are insufficiently detailed. A whole galaxy of failures are hidden with the blanket "SESSION NOT BOUND" or "APPLID PARAMETER INVALID".

User communities are highly variable. For example, a message text which is suitable for a group of engineers using APL may not be suitable for clerks using a CICS order entry system. Some of the other features recommended allow adding additional messages to convey problems in the network beyond the supplied capabilities of the product.

In a company with geographically dispersed operations it is desireable to be able to issue error messages in the predominant language within a region, for example French in Quebec or France, Spanish in Mexico, etc. This should be tied to the language of the presentation services map displayed at the terminal.

Suggested Implementation

A parameter deck which is read in either from a sequential file, a member of a PDS, or perhaps even a VSAM file. Assembling a deck of message definition macros, as is done with VTAM USS at the present time, is too inflexible.





vi) HELP!!!

An easily useable help facility should be provided. This would explain error messages which had just been displayed, or give guidance to the novice user on how to get logged on. It should be invokeable by:

1) Pressing PF1, or entering HELP or ? after an error message has been displayed. This would result in additional explanation of the message.
2) typing HELP or ? in any input field. The guidance supplied in this case should explain the use of the particular field.

If HELP or ? is entered in the application name field, or if PF1 is pressed when there is no abnormal condition, then a general help menu should be displayed.

The installation should be able to modify and add help information.

Justification

This should minimize calls to the network Hotline to obtain explanations of minor problems. The help information must be in the predominant language of the users of the system. User exit or screen layout map changes would require altering help guidance information.

Suggested Implementation

A consensus on the design of full screen help appears to be very close, with the HELP systems of SPF, DMS, and 8100 DPS being very similar. This design would appear, from its popularity, to be the best choice for the session control product.

vii) Broadcast News

The purpose of Broadcast News is to deliver non-application-specific information to the users of the network. This might include hours of service, or explanations of anticipated outages. There are three classes of News.

1) general news, which might contain extensive text. As an example, a list of dial access numbers. This information should only be displayed upon request from the terminal user.
2) serious news. This is a short item that should be immediately broadcast to all users of the network who are not in an application.
3) emergency news. This is a short item which must go immediately to all users of the network even if they are currently in an application. For example a warning that the network is coming down almost immediately.

Justification

A friendly, easily accessable, system is required for communicating information from central administration to all users of a network. Existing systems are all product specific. There is TSO broadcast and equivalent functions on most other data communications products, but no way to send information to everyone, for example about hours of availability of the network.





Suggested Implementation

By entering NEWS, or some equivalent, the user should be switched over to a news menu from which further information about the system or network which has been made available by the administrator can be obtained. The news may be browsed by the user, broken down by administrator selected categories which will be displayed on the first NEWS menu. When the user has selected a category he/she will receive a list of "headlines" in date order, with the most recent item at the top. The user may then select an item, either by typing a line number, tabbing down and typing a selection character, or by a selector pen operation. The full text of the news item should then be displayed. If there is more than one page of headlines, or if the full text of a news item is more than a page, the user may browse back and forth using PFKs 7 and 8 or equivalent.

News items could be creatable using SPF. A utility program could then copy the sequential data set, PDS member, or equivalent, into the news data base. Each item will automatically have a creation date and a purge date.

The facilities provided by INFO/SYSTEM are almost ideal for this function. The performance of the system with a large number of users seems to be quite adequate even with a database of information much larger than is forseen for the session control news function. It is also reasonably easy to create new databases. The principal weakness is in the presentation services area which may not be suitable for average users.

Serious news may be sent on the session which the session control product has with the terminal.

Emergency news must be sent even when there is no LU-LU session. For this it would probably be necessary to use the SSCP-LU session to send text data. If the session control product is not using the SSCP-LU session for its normal processing, then it would be necessary to use functions equivalent to an extension of the VTAM CNM Interface.

b) Application Program Interface

i) Receiving the Session Establishment Request

The application program will continue to receive the session establishment request with its own logon exit, but with the session establishment request will now come standardized data representing the values supplied by the user, and possibly modified by the session control system and its exits. This data must be independent of the application which is receiving it. The information received must be able to include all parameters which the application would normally prompt for, for example:

1) userid
2) password
3) new password
4) account and sub-account
5) miscellaneous parameters dependent upon the application

If the application chooses to reject the session establishment request it should do so in such a way as to allow the session control interface to explain the rejection to the end user.

The subsystem should retain the userid, password, and other identification for use by lower level subsystems. It is desireable that all command oriented interactive systems provide a method by which a command or stack of commands can be passed with the session initiation request.

Justification

As described previously, users of a network find it annoying to be prompted twice for similar information. If the session control system can collect all the necessary information up front then the extra prompt can be avoided. This may also improve the single system image by to some extent hiding the differences between different applications' session establishment processes.

When an application rejects a session establishment request at present no indication of this is passed back to the initiator of the session. All that the end user is told is that the session establishment request failed. A sophisticated user, monitoring for the NOTIFY RU which clears the session request is even misinformed that the session was rejected by the SSCP of the PLU. There is no provision for the application to specify a reason for rejecting the session. Valid reasons include invalid bind parameters, resources not available, terminal not defined in a table, invalid userid, etc.





If the applications accepted a command, or stack of commands it would be easier to provide to the end user the ability to directly enter applications which run under major sub-systems such as TSO or CICS.

Suggested Implementation

The only applications which currently accept data with the session establishment request are TSO and VCNA. The VCNA accepted data is syntactically similar to the TSO data. It is therefore suggested that a superset of the TSO LOGON command be used as the format for data passed to an application. Slightly more efficient might be a change to the CINIT RU to provide a formatted user data field.

The application program should be able to specify a sense code on the CLSDST OPTCD=RELEASE which is used to reject the pending session establishment request. This sense code should then be placed in the NOTIFY which goes to the session control interface which can then interpret the sense code for an error message to the terminal user.

If the application abnormally terminates, either during the logon or during normal processing, in such a way as to drop the terminal sessions without notification, it is desireable that a message explaining the abnormal termination be sent by the session control interface.

viii) Logical Systems

Some applications run as general purpose service suppliers to users, for example: TSO, CMS, or JES2. For a particular user coming into the network it is not particularly important which system his session is eventually allocated to, so long as the user's data is available. One mechanism, although not necessarily the only one, which can arrange such access is what might be called the "logical system". This is by extension of the older concept where a user had to be allocated to a particular physical system because of data integrity and other considerations.

There may be a one-to-one, one-to-many, or many-to-one relationship between logical and physical systems. If the device connections and performance permit it the one-to-many is perhaps the most popular. In this a single logical system or logical application, for example "TSO" is spread across a number of physical systems. But in this case a mechanism must be provided to choose which of the available processors a session will be allocated to.

Justification

If users are allowed to control their own choice of physical system, as is effectively required by the existing access method implementations, performance problems may arise. There may also be a requirement to control the number of simultaneous accesses by particular departments or groups of users to a particular application.

Suggested Implementation

Where there is a one-to-many situation two algorithms for selecting a system should be available. One is to balance the number of users between the various systems such that the ratio of the current number of users to the maximum number, as specified by the F TSO,USERMAX command, is as equal as possible.

The other is to fill up the first system in the list to its usermax before allowing any users to go to the second system, and so on.

A mixture of these options must also be allowed plus the option of writing a user exit to provide a local implementation. The choice of algorithms should be controlled by the Modify Application command discussed below.





c) Operation

i) Command Interface and Logging of Commands/Messages

All commands to the session control product should go through a common interface with the access method commands. Commands and Messages to the session control interface should be logged to the same facility as the commands and messages for other network operations functions. It should be possible to enter these commands either through a dedicated network operations screen, or through a shared operating system console.

The following list of commands is an indication of the general level of functionality required as extensions to the capability available now for operating the network. Some of these functions could be implemented in other components, such as a configuration management product.

Justification

Session control is the most important part of the service provided by an advanced network facility such as SNA. If something goes wrong with session control it is imperative to be able to synchronize diagnostic messages from the access method with those from session control. It is also desireable that the same operator be able to control both the access method and the session control interface from the same terminal or console.

Suggested Implementation

Commands should have a syntax similar to that of existing access method commands since they will probably be issued by the same people.

Commands would only have an effect on the portion of the session control product to which they were directed, just as VTAM or TCAM commands only affect the domain in which they are issued. Presumeably if more global results are desired, a product such as NCCF could be employed. Many of the commands will only be truly effective in a Configuration Management Configuration.

ii) Commands

The following capabilities are desireable:

D-1 Display Host

displays active applications, active userids by application

D-2 Display Application

Note that this is not the VTAM application id, but rather an installation defined name, possibly used by end users when specifying an application. An example might be TSO where this represents TSO on more than one system in a complex.

displays host(s) on which the application is running, active userids, secondary logical units in session, the owning Host for each secondary logical unit.

D-3 Display Userid

displays the Host(s) active on, the application(s) active on, the secondary logical unit name(s), and the owning Host(s) for the SLU(s).



Justification

The VTAM display tsouser command will not display a user on CICS, IMS, etc. Neither will the MVS D TS,L command.

D-4 Display secondary logical unit

displays Host connected to, Host which owns it, application connected to, userid active on.

V-1 Drain or Resume a Host

If a Host is draining or drained, the session controller will reject attempts to establish a session with any applications on that Host. Resume reverses the effect of Drain. This would be preparatory to taking the host system out of the network, or shutting it down.

V-2 Drain or Resume an Application

If an application is draining or drained, the session controller will reject attempts to log on to the application. This would be a general capability similar to the F TCAS,USERMAX=0 command to prepare for



taking an application down.

V-3 Drain or Resume a Host for a Specific Application

If a particular host is drained with respect to a specific application then the session controller will reject attempts to log on to the application on the particular host. This might be used to reduce the load from a particular application on a particular host system, so as to improve performance for the remaining users.

S-1 Start or Stop a Host

S-2 Start or Stop an Application

S-3 Start or Stop a secondary logical unit

If the secondary logical unit is stopped, the session control interface will break its LU-LU session with the SLU and no session establishment requests will be accepted.

F-1 Modify Application

You may alter: the list of Hosts on which the application is running, and the algorithm for choosing between them, and the maximum number of users per host.

F-2 Modify logical unit

Used to control the access from that particular logical unit to particular Hosts, or particular applications.

Justification

The need to control a more abstract concept than a logical unit and to obtain information about the status of sessions. Current VTAM/TCAM displays and commands are inadequate. Commands are required to control logical systems and load balancing.

Suggested Implementation

Commands should follow a VTAM like syntax with D (or DISPLAY), V (or VARY), S (or START), P (or STOP), and F (or MODIFY). It may be desireable not to use NET as the first positional operand. In that case, if the commands are not to be routed via access method command facilities, they should be distinguished by NCCF.

d) Accounting Interface

The session control interface will allow for an accurate correlation between network accounting information, which tends to be terminal oriented, and application accounting information, which tends to be user oriented. To account for the use of network resources, and to assist the application in accounting for the use of its resources, the session control product must accurately identify the user, via passwords, identification cards, voiceprints, retinal scanners or whatever is available. The terminal involved must also be accurately identified along with its characteristics. The session control interface will not do the accounting, just provide the accounting information.

Justification

As the cost and complexity of network communications increases, while the cost of host processing declines, it becomes more essential to be able to accurately assign the networking costs to the users of the service. Since a single terminal may be connected to a wide variety of services during a day, some of which, such as message switching, will be almost entirely networking functions, it is necessary to associate the usage of the network with particular users. In a lot of current systems, for example, charging for the use of dial ports is done by the applications. if you add another application, which with VTAM takes a matter of seconds, you must duplicate the dial charging.

In a complex network there are a large number of resources which are not accounted for by conventional systems which are application based. For example, TSO does not have access to information about the physical path from the terminal which the data traversed, but this affects the cost to the company of delivering that TSO session. The cost of the network resources should be billable to the user.

Suggested Implementation

Session control should pass the user identification and terminal identification and characteristics to the target application to permit that application to perform its own accounting, and, in order to allow for network utilization measurements, it should call a user exit with the accounting information. The exit should be called:

1) When the LU establishes a session with the session controller.
2) When the LU has been successfully/unsuccessfully passed to an application.
3) When the LU returns from the application.
4) If the security exit or RACF rejects a session initiation





request.

Some examples of data which should be available at the exit are:

1) SLU name
2) actual application name (PLU name)
3) logical application name (generalized TSO for example)
4) network userid
5) date and time
6) subarea number of the PLU
7) subarea number of the SLU
8) subarea number of the session controller
9) class of service

e) Security

The application is responsible for performing function and data access security checking based upon the accurate network identification of the user and the originating source terminal. The security responsibility of session control should be limited to restricting the initiation of sessions.

Session control should reject the session request if told to do so either by the optional user exit discussed in section 1.a.iii above, or by RACF. Control of the rejection should lie as close to the end user department as possible. If possible it should not depend upon central site system programmers. It is not the session controller which makes the decision to reject the session initiation request.

Justification

There is too much variation in requirements between companies, and even between departments within the same company for a single security algorithm to work. However, it is desireable for the user to be informed of the rejection as promptly as possible, and if possible without wasting time passing the request around the network. For example, at present, in order to reject a TSO logon for security reasons it is necessary to go as far as creating the TSO address space, which is then immediately shut down. The only use of the address space is to send the rejection message to the user.

Management of the various departments has the responsibility for controlling access and should have as much direct control as possible.

Suggested Implementation

It should be possible for the RACF administrator to specify a verification exit by RACF user, group, or super-group. The end-user cannot change this specification. The exit should be invoked by name from a library of exits. The exits should be implementable in common higher level languages such as COBOL, FORTRAN, or PL/I. This should permit the department's RACF administrator to implement the verification function without recourse to outside resources.





f) Recovery

In the event of a shutdown of the Session Control interface, either planned or unplanned, it is necessary to retain some information for later restart. When restarting the session control interface after the outage, the operator should be able to specify whether the restart is to be WARM, COLD, or FORMAT.

A "FORMAT" start should cause the product to establish the initial operating environment. The files recording the status are formatted according to information from customization information.

A "COLD" start reinitializes status to an initial condition in which all information on already established sessions is lost, but information on the configuration of the network and logical resources in it is retained. The tables controlling where new sessions are to be routed are reinitialized from customization information.

A "WARM" start attempts to reestablish the environment prior to the outage. Sessions which were in existence prior to the outage are reestablished subject to security limitations.

The recovery information contains the terminals which were currently being supervised by the session control interface and their status, authority and user selected options. It includes information on existing sessions at the userid level. It also contains the status of the generic applications and the Hosts.

Justification

After short outages of the network it should not be necessary for hundreds, or even thousands of users to reenter a logon. Their sessions should be reestablished automatically. For longer outages there is a security risk in reestablishing sessions, and the network should be reset to an initial condition.

A user should be able to logon reconnect to an application such as TSO without having to worry about what system the disconnected address space is running on. Possibly the user should not even have to specify that a reconnect is required. If the userid is still on a system somewhere, then the session should be reestablished.

Some terminals do not have a mechanism for generating a logon, for example 3270 printers. There is a need for an ability to automatically reestablish printer sessions after a network outage.

Suggested Implementation

There are three categories of information to be recorded here. Customization information, such as panel layouts, table sizes, and application identifiers could come from an initialization deck, or PARMLIB member. Long term information includes the identifications and attributes, including account ownership, of resources in the network. Typical resources for this purpose are userids, terminals, logical applications, and systems. This may require a different recording mechanism from the short term, volatile information such as what system a logical application is running on; to what application, and from what terminal, is a particular userid connected; and so on. The long term information might be recorded in a database of some sort. The volatile information might be better recorded on a log file which would be sequentially read during a warm start. Customization information, such as panel layouts, table sizes, and application identifiers could come from an initialization deck, or PARMLIB member.





Consolidated Glossary

| | |
|---|---|
| display terminal | - any logical unit which communicates with the end user through a rectangular display surface while performing only presentation services mapping to the display surface, and which has the ability for the application program to control the placement of output data on the presentation space, and to determine what fields on the presentation space were modified by the terminal user. Examples include all 3270 terminals, the 8775, and terminals supported through NTO which have cursor addressable displays such as the IBM 3101. |
| non-display terminal | - any logical unit which is capable of initiating a session request and which performs only presentation services mapping to the terminal end-user, but which does not have the capabilities specified for a display terminal. Examples are the 3767, 377x, the 2741, and the ASR 33/35. |
| colour | - Canadian for color |
| selector pen | - also known as a light pen. This includes all mechanisms used merely to indicate that a particular field has been selected by the end-user. |
| end-user | - the person requesting services from the network and computer systems. |
| administrator | - the person within a user department responsible for tailoring the session control interface to the department's requirements. |