

Dave Lyons  
Digital Equipment Corporation  
March, 1981

The information in this document is subject to change without notice and should be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license.

**The DECSYSTEM-20 TCP/IP user interface**

Digital Equipment Corporation assumes no responsibility for the use or reliability of its software on equipment that is not supplied by DIGITAL.

Copyright (C) 1981 by Digital Equipment Corporation

The following are trademarks of Digital Equipment Corporation:

This document describes the USER / TCP interface for the DECSYSTEM-20 operating system. It also includes information on an interface which may be used for developing experimental higher level protocols.

DIGITAL  
DEC  
PDP  
DECUS  
UNIBUS  
COMPUTER LABS  
CONTEXT  
DOL  
DECCON

LOCAL  
INDAC  
LAB-8  
DECSYSTEM-20  
TYPSET-8  
TYPSET-10  
TYPSET-11

The information in this document is subject to change without notice and should not be construed as a commitment by Digital Equipment Corporation. Digital Equipment Corporation assumes no responsibility for any errors that may appear in this document.

The software described in this document is furnished under a license and may be used or copied only in accordance with the terms of such license.

Digital Equipment Corporation assumes no responsibility for the use or reliability of its software on equipment that is not supplied by DIGITAL.

Copyright (C) 1981 by Digital Equipment Corporation

The following are trademarks of Digital Equipment Corporation:

|               |              |            |
|---------------|--------------|------------|
| DIGITAL       | DECsystem-10 | MASSBUS    |
| DEC           | DECTape      | OMNIBUS    |
| PDP           | DIBOL        | OS/8       |
| DECUS         | EDUSYSTEM    | PHA        |
| UNIBUS        | FLIP CHIP    | RSTS       |
| COMPUTER LABS | FOCAL        | RSX        |
| COMTEX        | INDAC        | TYPESET-8  |
| DDT           | LAB-8        | TYPESET-10 |
| DECCOM        | DECSYSTEM-20 | TYPESET-11 |

This document describes the formats and relations of the USER interface and the Department of Defense Transmission Control Protocol and Internet Protocol.

This document is divided into four sections. The first deals with the "Raw packet" interface to the Internet Protocol. This interface may be used by programmer to implement higher level protocols such as the Transmission Control Protocol, a datagram service, or a specialized protocol such as voice transmission. This interface is not intended to be used as a user datagram service. It is included to allow protocols other than TCP to be implemented and used.

The second section is concerned with the TCP / USER interface itself. It describes the various attributes that may be associated with a connection, and the formats of the monitor calls to set and read these attributes. It is assumed that the reader is familiar with the DECSYSTEM-20 user interface for files.

The third section explains the interrelations of the user interface and certain network events (such as receiving data). This section also covers differences between "local" I/O, and "network" I/O.

The fourth section covers the retransmission algorithm exponential smoothing function.

## Section 1. The INTERNET interface.

## IPOP% JSYS (754) - I P operations

This section covers the IPOP% JSYS and its functions. These functions can be used to send and receive special messages directly, or to set various system parameters related to the IP network layer.

Restrictions: Some functions may require WHEEL or OPERATOR. In addition, some other functions may require NETWORK-WIZARD.

Accepts in AC1: JFN, Special Queue number, or other designator.  
 AC2: Function code and optional flags.  
 AC3: Argument or pointer to argument block.

Returns: +1 Always

A RESET% JSYS will release all special queues that have been assigned.

| Code | Symbol         | Meaning  |
|------|----------------|--|
| 0    | .IPASQ         | Assign special queue for IP packets. This function requires NETWORK-WIZARD. AC3 points to a block of the form: |
|      | Word 0: .IPPPV | ! 32 bits internet protocol ! 4 bits !<br>! number ! !   |
|      | Word 1: .IPFHV | ! 32 bits foreign host ! 4 bits !<br>! number ! !  |
|      | Word 2: .IPLHV | ! 32 bits local host number ! 4 bits !   |
|      | Word 3: .IPPTV | ! 16 bits local port, ! 4 bits !<br>! 16 bits distant port ! !   |
|      | Word 4: .IPPPM | ! mask for .IPPPV ! 4 bits !   |
|      | Word 5: .IPFHM | ! mask for .IPFHV ! 4 bits !   |
|      | Word 6: .IPLHM | ! mask for .IPLHV ! 4 bits !   |
|      | Word 7: .IPPTM | ! mask for .IPPTV ! 4 bits !   |

If IP%SPP is set, word 3 distant port bits are ignored.

Other flag bits in AC2 are:  
 IP%RPI = 1B0 ;Raw Packet Interface  
 IP%SPP = 1B1 ;Single Port Protocol

On a successful return, AC1 contains a Special queue handle.

- 1 .IPRSQ Release IP special queues. AC3 contains the queue handle or -1 for all assigned queues.
- 2 .IPSSM Send Special Message. AC3 points to an argument block of the following form.
1. Word 0 of the buffer must be a word count (including word 0)
  2. Words 1 thru 5 must be a legal Internet Header
  3. If IP%SPP was set when the ASNSQ jsys was done, word 6 must contain port number(s)
- The monitor will validate the local host field and compute the header checksum.
- 3 .IPRSM Receive Special Message. AC3 points to a buffer which receive the data. The first word of the buffer must contain the length of the buffer in the right half. After the call, the length is stored in the left half. If the message is too long, a size error is given, and the message is truncated to the length of the buffer.

## Possible errors:

- NTWZX1 NETWORK-WIZARD capability required
- IPOX1 Illegal function code
- IPOX2 All Special Queues are in use
- IPOX3 Special Queue allocation conflict
- IPOX4 Invalid message size
- IPOX5 Insufficient system resources
- IPOX6 Invalid header value in this queue
- IPOX7 Queue handle out of range
- IPOX8 Queue was not assigned
- IPOX9 WHEEL, OPERATOR or NETWORK-WIZARD capability required

## Section 2. The USER / TCP interface

It is the programmers responsibility to provide information to the operating system about the network connection required. This information is passed to the operating system thru a GTJFN monitor call. Although TCP has full duplex data connections, there is a need for two types of connections, active, and passive. The active connection can be thought of as going out and finding someone to talk with. This is also referred to as the Caller. The passive connection can be thought of as waiting until someone shows up to talk with. This is also referred to as a Listener. The format used by the TCP network protocol is as follows.

```
TCP:[LOCAL<HOST][ -LOCAL<PORT[#]].[FOREIGN<HOST][ -FOREIGN<PORT][;A1..]
```

The following examples show the various ways that connections may be specified.

1. TCP:.RADC-TOPS20-1;CONN:ACTIVE
2. TCP:1#
3. TCP:1#.-5000
4. TCP:1#.1200200002;CONN:PASSIVE
5. TCP:4500000254.1200200117-3;CONN:ACT
6. TCP:4500000254-177#.1200200117-123;CON:A;BUF:128;PER:0  
;TI:60;TY:2;SEC:2;COM:2

In case 1, the local information is not present and will be defaulted. Port numbers are controlled as follows:

If the number is in the range of 0 to 377 (octal), it must be followed by a "#", and the user must have WHEEL, OPERATOR or NETWORK-WIZARD. The "#" is redundant, but is included to prevent mistakes. This range of numbers is reserved for system servers.

The numbers between 400 and 77777 are reserved for users and are not controlled by the system.

Numbers in the range of 100000 to 177777 are reserved by the system, and assigned on a "need" basis.

Note: The monitor will assign the numbers in the following manner. The "sign" bit of the port number will always be set, the next 9 bits will be reserved for the job number, and the last 6 bits will be the the JFN number assigned the

connection. These assignments are subject to change without notice.

As example 1 fits into the third case, it will be assigned a port number by the system. The destination system is identified as "RADC-TOPS20" and the port number is "1". The CONNECTION attribute is present with the ACTIVE option, and this system will attempt the connection.

The name/Internet Number binding is local to only this system. It is the responsibility of the system administrator to maintain this binding. There are no restrictions on having many names identify the same system, or having the same name on different systems disagree. The later situation should be avoided, but this is not required.

The second case shows a passive open on port 1. The connection will be accepted from any host and any port

The third case shows a passive open on port 1 that will only accept connections from any system with a port number of 5000.

The fourth case shows a passive open on port 1 for any user on node 1200200002 (SRI-KL).

The fifth case shows an active connection from host 4500000254 (the DECnet side of DEC-2136) with a defaulted port number. The target side of the connection is the ARPANET side of DEC-2136.

The sixth example shows a connection from DEC-2136 (DECnet) to DEC-2136 (ARPANET). This connection includes the following attributes:

```

;CON:A show an active connection
;BUF:128 buffer size of 128 bytes
;PER:0 persist in opening the connection
;TI:60 time the connection out if there is a 60 second
failure
;TY:2 type of service is Speed vs. Reliability
;S:2 security level of 2
;COM:2 compartment of 2

```

The following are valid attributes:

**;CONNECTION:ACTIVE**

**;CONNECTION:PASSIVE** This attribute is used to indicate if the operating system should actively connect to the foreign system. The default is PASSIVE.

**;BUFFER:n** This attribute indicates that record mode I/O with records of N bytes is to be used. N may be in the range of 0 to  $2^{16}-1$ . If not given, unbuffered I/O will be used.

**;PERSIST:n**

**;PERSIST:(n,m)** With an argument of 0, attempt to open the connection and keep trying until successful. If n is given, try for n seconds, at which time an error return is given if no connection has been established. An attempt is made every m seconds, where m is an estimate of the round trip delay between systems or provided by the user. If no persistence is given, 30 seconds will be used.

**;TIMEOUT:n** The amount of time allowed to pass while waiting for a message from the foreign system before an error is given to the user. If not given, a default 30 seconds will be used. If the value of n is 0, no timeout will occur. N may be in the range of 0 to  $2^{18}-1$ .

**;TYPE-OF-SERVICE:n** The type of service required by the user indicates what tradeoffs are to be made in providing data transmission. N may be in the range of 0 to  $(2^{18}-1)$ . The TCP implementation will only use the low order 8 bits.

**;SECURITY:n** The security level of the connection may range from 0 to 3. If not specified, a value of 0 will be used.

**;COMPARTMENT:n** The compartment level of a connection is specified as a code provided for the user by the Defense Communications Agency. N may be in the range of 0 to 255.

Note that these last two fields are not used in TCP itself, but in lower layers of the protocols. They are provided for at this level to allow the user control over these fields. Also, the use of either of these two fields will invoke a request to the Access Control Job, or will fail if the Access Control is not enabled for these functions. The TYPE-OF-SERVICE field will also invoke a call to the Access Control Job, but will be allowed if Access Control is not enabled for

this function.

TCOPRZ 32YS (788)

The following are the TCOPR functions related to the TCP: device. These functions are like the MTOPR 32YS, but are only used with the TCP device.

ACCEPTS AC1: JFR of device  
AC2: function code (see below)  
AC3: function argument or address of argument block.  
See each function for details.

RETURNS +1 always

Code Symbol Meaning  
0 .TCRCS read connection status. ACS points to a block which at least .TCRCL words long.

|   |        |   |
|---|--------|---|
| 0 | .TCLEN | Length of block                             |
| 1 | .TCFTR | foreign port<br>18 bits, right justified    |
| 2 | .TCFTH | foreign host<br>32 bits, right justified    |
| 3 | .TCFLP | local port<br>18 bits, right justified      |
| 4 | .TCFLH | local host<br>32 bits, right justified      |
| 5 | .TCRWR | Receive window<br>18 bits, right justified  |
| 6 | .TCRWS | Send window<br>18 bits, right justified     |
| 7 | .TCRCS | Connection state<br>4 bits, right justified |

|        |              |
|--------|--------------|
| .TCRCL | closed       |
| .TCRPI | listen       |
| .TCRPS | syn sent     |
| .TCRPR | syn received |
| .TCRES | established  |
| .TCRF1 | fin wait 1   |
| .TCRF2 | fin wait 2   |
| .TCRTW | time wait    |
| .TCRCW | close wait   |
| .TCRCG | closing      |

## TCOPR% JSYS (755)

The following are the TCOPR functions related to the TCP: device. These functions are like the MTOPR JSYS, but are only used with the TCP device.

ACCEPTS AC1: JFN of device  
 AC2: function code (see below)  
 AC3: function argument or address of argument block.  
 See each function for details.

RETURNS +1 always

| Code | Symbol             | Meaning   |
|------|--------------------|---|
| 0    | .TCRCS             | read connection status. AC3 points to a block which at least .TCRCL words long. |
|      | .TCLEN 0           | Length of block   |
|      | .TCTFP 1<br>TC%TFP | Foreign port<br>16 bits, right justified  |
|      | .TCTFH 2<br>TC%TFH | Foreign host<br>32 bits, right justified  |
|      | .TCTLP 3<br>TC%TLP | Local port<br>16 bits, right justified  |
|      | .TCTLH 4<br>TC%TLH | Local host<br>32 bits, right justified  |
|      | .TCTRW 5<br>TC%TRW | Receive window<br>16 bits, right justified                                      |
|      | .TCTSW 6<br>TC%TSW | Send window<br>16 bits, right justified   |
|      | .TCTCS 7<br>TC%TCS | Connection state<br>4 bits, right justified                                     |
|      | .TCPCL             | closed  |
|      | .TCPLI             | listen  |
|      | .TCPSS             | syn sent  |
|      | .TCPSR             | syn received  |
|      | .TCPES             | established   |
|      | .TCPF1             | fin wait 1  |
|      | .TCPF2             | fin wait 2  |
|      | .TCPTW             | time wait   |
|      | .TCPCW             | close wait  |
|      | .TCPCG             | closing   |

.TCTBW 10      Buffers waiting ack  
 TC%TBW            8 bits, right justified

.TCTBP 11      Buffers pending receipt  
 TC%TBP            8 bits, right justified

.TCTBS 12      Buffer size  
 TC%TBS            16 bits, right justified

.TCTTS 13      Type-of-service and security fields  
 TC%TTS            18 bits, 1st 18 bit byte  
 TC%TSF            2 bits, 3rd 9 bit byte  
 TC%TCF            8 bits, 4th 9 bit byte

The bits used in the TCP implementation are as follows.

.TCTPR 3 bits    Precedence  
 .TCTST 1 bit    Stream / Datagram  
 .TCTRE 2 bits   Reliability  
 .TCTSR 1 bit    Speed / Reliability  
 .TCTSP 1 bit    Speed

.TCTTT 14      Transmission timeout  
 TC%TTT            9 bits, right justified

.TCTUD 15      Urgent data information (to be defined)

.TCTRA 16      Retransmission parameters - Alpha  
 TC%TRA Alpha, a floating point number.

.TCTRB 17      Retransmission parameters - Beta  
 TC%TRB Beta, a floating point number.

See the section on message retransmission for a description of these fields, their uses, and limitations.

.TCTPI 20      PSI channel assignment. See function .TCSPC for a definition of this field.

1 .TCSUD            Send urgent data. AC3 points to a block of the form

0      Pointer to data  
 1      Count of bytes or 0  
 2      Byte to terminate output on

(note that these are the same as AC2 - AC4 of the SOUT and SOUTR JSYS's)

Setting TC%FEL in AC3 will force End Of Letter to be set.

- 2 .TCSBS Set buffer size. AC3 contains a number in the range of 0 to  $2^{16}-1$ . A value of 0 indicates non-buffered mode.
- 3 .TCSPA Set passive/active flag. TC%APF is set in AC3 to indicate an active connection, and cleared to indicate a passive connection.
- 4 .TCSPP Set persistence parameters. AC3 is the time to wait for connections.
- If AC3 is 0, do not timeout the connection.
- If AC3 contains 0,,n an attempt to connect will be made for n seconds with retransmission time estimated by the system
- If AC3 contains m,,n an attempt to connect will be made for n seconds with retransmission every m seconds. M must be less than n.
- 5 .TCSTP Set timeout parameters. AC3 contains the time to wait before a timeout. The value given must be in the range of 0 to  $2^{18}-1$ .
- If AC3 contains a 0, no timeout will occur.
- 6 .TCSRP Set retransmission parameters. AC3 points to an argument block which is two words long. The first word contain a floating point number, Alpha, and the second contains another floating point number Beta. For a description of these two fields see the section on message retransmission.
- 7 .TCSTS Set Type-of-service. AC3 contains the type of service desired. The value must be in the range of 0 to  $2^{18}-1$ .
- 10 .TCSSC Set security and compartment levels. AC3 contains the security level and the compartment levels in the form (security),,(compartment)

Security may be in the range of 0 to 3.  
Compartment may be in the range of 0 to 2+8-1.

- 11 .TCSPC Set PSI channels. AC3 contains six 6 bit fields as follows
- |        |                                      |
|--------|--------------------------------------|
| TC%TPU | 1st byte, Urgent data channel        |
| TC%TRI | 2nd byte, Read data channel          |
| TC%TSI | 3rd byte, Send data channel          |
| TC%TER | 4th byte, Error channel              |
| TC%TSC | 5th byte, State change               |
| TC%TXX | 6th byte, unused, must be 77 (octal) |
- To indicate that no interrupt is desired for a given function, specify the value 77 (octal) for the channel.
- 12 .TCRTW Read a single entry from the TCB. AC3 contains the word of the TCB that is desired. On return, AC3 will contain the value of the word in question.
- For a list of the words that may be returned, see the function .TCRCS.
- 13 .TCSIL Set the interrupt level for buffers. AC3 contains a number between 0 and 1024 in each half. The left half sets the number of bytes received before an interrupt will occur. The right half sets the number of bytes which must be available in the output buffer before an interrupt will occur. In both cases, a value of 0 (the default) will be treated exactly like a value of 1.
- 14 .TCSSR Set the route to be used in transmission of the message. AC3 points to an argument block. The first word of that block is the total length of the block. Each word thereafter is the Internet Address of the next node to route to, right justified.
- 15 .TCRLB Read lower bound for retransmission. The number of seconds is returned in AC3 as a floating point number.
- 16 .TCSLB Set lower bound for retransmission. The number of seconds will be in AC3 as a floating point number. Requires WHEEL, OPERATOR or NETWORK-WIZARD. The number must be larger than 0 and less than the current

- upper bound.
- 17 .TCRUB Read upper bound for retransmission. The number of seconds is returned in AC3 as a floating point number.
- 20 .TCSUB Set upper bound for retransmission. The number of seconds will be in AC3 as a floating point number. Requires WHEEL, OPERATOR or NETWORK-WIZARD. The number must be larger than the current lower bound and less than 250.

The following functions do not require a JFN in AC1, as they are system wide. AC1 must contain a 0.

- 100 .TCRDLB Read default lower bound for retransmission. The number of seconds is returned in AC3 as a floating point number.
- 101 .TCSDLB Set default lower bound for retransmission. The number of seconds will be in AC3 as a floating point number. Requires WHEEL, OPERATOR or NETWORK-WIZARD. The number must be larger than 0 and less than the current upper bound.
- 102 .TCRDUB Read default upper bound for retransmission. The number of seconds is returned in AC3 as a floating point number.
- 103 .TCSDUB Set default upper bound for retransmission. The number of seconds will be in AC3 as a floating point number. Requires WHEEL, OPERATOR or NETWORK-WIZARD. The number must be larger than the current lower bound and less than 250.

Error codes:

- DESX1 Invalid source/destination designator
- DESX3 JFN is not assigned
- DESX4 Invalid use of terminal designator or string pointer
- DESX9 Invalid operation for this device
- IOX5 Device or data error



### Section 3. I/O interrelations.

This section describes the relations of various monitor calls which relate to performing I/O for a TCP device.

First, the user may execute a TCOPR JSYS to set various fields within the transmission control block (TCB). Some of these functions are allowed only before the OPENF. These include

- .TCSSC Set security and compartment levels
- .TCSPA Set passive/active flag (makes no sense after the OPENF has been done)
- .TCSBS Set buffer size
- .TCSPP Set persistence parameters (makes no sense after the OPENF has been done)

If the user has enabled for PSI interrupts on state changes, an interrupt will be given the user every time the state machine is stepped.

If the user has enabled interrupts for URGENT data, an interrupt will be given to the user each time the urgent data pointer is increased, that is, we have been notified of more urgent data than we previously knew about. If the user has not read all of the previous urgent data, the interrupt will still be sent.

If the user has enabled interrupts for errors, a PSI interrupt will be given under the following conditions:

1. A timeout has occurred (either in communications, or in opening the channel).
2. An ABORT or RST (reset) message was received.

These types of events also step the state machine, and will also cause an interrupt on the state channel.

SIN, SINR, BIN, SIBE, DIBE and input interrupts.

If the user has enabled for input data interrupts, the program will be interrupted when the amount of unread user data exceeds the value provided with the .TCSIL function or non-zero if no value was specified. At any time, a SIBE will return the number of unread bytes available to the user, i.e. those bytes which have been acknowledged. This is not the receive window size. To determine the size of the receive window, the program must use a TCOPR JSYS. The actions of the SINR JSYS are documented in the monitor calls manual, and the TCP flag "EOL" (end of letter) will be used to determine record blocking.

SOUT, SOUTR, BOUT, SOBE, SOBF, DOBE, and output interrupts.

If the user has performed a passive open, and then executes a JSYS which will cause output to occur, the TCB is converted to an active open, and the user will block until the connection is established and the data specified by the JSYS is moved into the monitor, or an event which will cause the user to be notified of an error occurs.

If the user has enabled for output data interrupts, the program will be interrupted when the output buffer becomes "non-full", that is, space is available in the output buffer. If the user has indicated a "buffer level" with the .TCSIL function of the TCOPR JSYS, the interrupt will be signaled when the free space in the buffer exceeds the specified level. At any time, a SOBE or SOBF will return the number of bytes which are in the output buffer, i.e. the number of bytes which have not been acknowledged by the foreign host. When a SOUTR is executed, the data message sent to the foreign system will have the EOL bit set, and the byte numbering will be adjusted to the next record boundary. This record size is set with the ;BUFFER attribute to GTJFN, or with the .TCSBS function of the TCOPR JSYS.

#### Single Fork I/O. A short description.

Some concern over how to perform I/O to multiple connections with a single fork have been raised. The way to do this is fairly straight forward. The following description demonstrates how this might be accomplished. For simplicity, we will only examine the case of a single input stream, and assume some critical background process (like computing pi to 100,000,000 places). We will also assume that the program is only willing to be interrupted when 100 or more bytes are available.

1. The program opens a connection to the source of the data.
2. The program enables interrupts for receive data, and sets the interrupt level to 100. This is done with the TCOPR functions .TCSPC and .TCSIL.
3. The program starts computing pi.
4. At some point in time, an interrupt occurs, and the calculation of pi is interrupted. At this point the programmer knows there has been at least 100 bytes of data available since the last interrupt. The program may read any amount of this data. Let us look at the implications.
  1. The fact that an interrupt has occurred implies that the condition for the interrupt has been true at least once since the last interrupt. This does not guaranty that the condition is still true. The program must assure that these conditions still exist within the interrupt code. This is true for ALL of TOPS-20.

2. In light of number 1, if a program receives an interrupt for data available, and reads data until less than 100 bytes remain, that program will receive an interrupt for data when more than 100 bytes are available, regardless of when this event occurs.
3. Assume the program receives an interrupt for data, and 120 bytes are available. The program then reads 60 bytes. While the program is processing this data, another 45 bytes of data is received. The program checks for data, and finds that 105 bytes are available. The program reads all the data, and will then dismiss the interrupt. As the conditions in number 1 are true, the program will receive another interrupt for data being available right away, even though there is no data available. As number 1 points out, the program must assure that the conditions are still true. In this case, they are not, and the program should dismiss the interrupt.
4. If the program receives an interrupt, fails to read enough data to drive the number of bytes below the current alarm level, and dismisses the interrupt, no more interrupts will be given for that channel. There can be no transition over the alarm level.
5. The program enters a loop to read the data. First, the program executes a SIBE JSYS to determine the exact number of bytes available. The program then reads this data, and processes it. The program continues in this loop until the SIBE JSYS indicated there is no more data to be read.
6. The program dismisses the interrupt, and the calculation of pi continues.
7. The program completes the calculation of pi, prints the results, closes the TCP connection, and exits.

It is probably easiest to think of interrupts as a "pulse" which will be queued for a job until that job is willing to receive it (i.e. at a lower interrupt level, or not at interrupt level at all). The "pulse" will remain queued even if the conditions which caused it no longer true. Many different interrupts may be queued for a job, and they will be processed in order of occurrence within order of level. Again, this is true for all of TOPS-20.

#### Section 4. Data retransmission

This section describes the data retransmission parameters, Alpha and Beta.

The functions used provide an "elongation factor" or delay variance, Alpha, as well as an exponential smoothing function with weighting of Beta. Assume the following:

RTT is the delay between the sending of an octet and the ACK for that octet.

SRTT is the exponentially weighted round trip delay.

UBND is the upper bound for data retransmission.

LBND is the lower bound for data retransmission.

RTO is the retransmission timeout for this data set.

SRTT may be calculated from the following.

$$SRTT = ( Beta * SRTT ) + ( ( 1 - Beta ) * RTT )$$

RTT and SRTT are initially set to an estimated delay time. The retransmission time out, RTO, may be calculated as follows:

$$RTO = \max ( LBND, \min ( UBND, Alpha * SRTT ) )$$

By default, the lower bound, LBND, will be set to 1 second and the upper bound, UBND will be set to 250 seconds. These values may be changed with the .TCSLB and .TCSUB functions of the TCOPR% JSYS. Beta must be in the range of 0.0 to 1.0. It will initially be set to 0.9. Alpha must be in the range of 0.0 to 250.0. The default value for Alpha is 1.5. Although numbers less than 1.0 and larger than 10.0 are fairly useless, they are allowed.

The system administrator may set the default upper and lower bounds on retransmission. This may be done with the .TCSDL and .TCSUL functions of the TCOPR% JSYS.

-----