Internet Experiment Note 108

Internet User Queues

William W. Plummer

Bolt Beranek and Newman, Inc.
50 Moulton Street
Cambridge MA   02138

19 June 1979

Internet User Queues

William W. Plummer 19 June 1979

The Internet JSYSes provide a facility for user programs to send and receive Internet packets. This set of calls is parallel to the "IMP special queue" calls, which dealt only with the ARPANET.

In order to use the Internet calls a program must first assign an Internet Queue using ASNIQ (JSYS 756). The arguments for this JSYS are a set of mask and value words that determine what messages may be sent and which of the incoming messages will be delivered to the owning job. A successful ASNIQ returns with a handle (a small number, short-hand for the queue), in T1 which can then be fed to SNDIN and RCVIN (SeND InterNet, ReCeiVe InterNet).

The arguments for SNDIN are an Internet Queue Handle and a pointer to a buffer of data in the caller's space. Word-0 of this buffer is the size of the buffer, including this count word. Words 1 through 5 must have a Version 4 Internet Header (see last page). Various fields of the header are checked for legality, the source host word is filled in, the Internet header checksum inserted. Then a network interface is selected and a local leader for that network is constructed in the packet. The packet is then queued for output on that interface and SNDIN returns.

Basically, users have no control over which interface is selected for packet output and no knowledge of which interface an incoming packet arrived on. This is analogous to the ARPANET where users have no control over which IMPs a message actually goes through. There is an exception when a system is equipped with a "Raw Packet Interface" (RPI) which is used to connect to a remote gateway or a encryption device (such as a BCR). Bit zero in T1 to ASNIQ will flag the queue as using the RPI rather than the normal interfaces.

RCVIN takes the same arguments, an Internet Queue Handle and a buffer pointer. Normally, if no messages are already waiting on the queue, RCVIN waits. This wait can be defeated by turning on the "don't wait" control bit which forces an error return if no messages are waiting. When a message is available, it is delivered to the user- supplied buffer. The number of words filled is set into the left half of word-0 (the count word) of the buffer. Ordinarily this can be ignored, but if the message was too big for the buffer, this tells how much space would have been required.

When finished, the queue handle can be released by using RELIQ. All handles owned by the job may be released by supplying -1 as the argument.

All Internet JSYS calls use only the left 32 bits of each word. This is true of both the ASNIQ argument block and of data buffers.

Messages left waiting on an input queue for a long time (currently 30 seconds) will be deleted. Also, a queue cannot hold very many messages (32, today). Flooding an Internet input queue to a slow receiver will result in dropped messages.

Messages addressed to the sending host will ordinarily be delivered without sending over any network at all and are reasonably fast.

User programs have no control over which network a given message will be sent out over. This decision will be made by the host gateway module on the basis of routing information supplied to it by other Internet gateways. This means that all networks to which a host is connected must go down before Internet communications will be completely stifled, and even then, forks within a job will be able to communicate due to the local delivery mechanism.

Programs using Internet messages must be aware that messages are not necessarily delivered in the order in which they were sent, some messages may be dropped and others duplicated. Some may traverse a broadcast network and be clobbered by other packets, lightning, flaky intermediate gateways, etc. Thus, some higher level protocol is needed in most cases. TCP is one example, but there are others (the "Datagram Protocol", XNET, etc.).

If a particular protocol is implemented in the monitor (e.g., TCP) and that protocol module is turned on, no messages of that protocol type will be passed to users via Internet queues. Assigning a queue will still be possible, but no traffic will reach the user unless that protocol module is disabled.

Not all Internet protocols have ports. If you are implementing one which does not, be sure that .IQPTM in the ASNIQ block contains a zero. If the protocol uses ports, they are expected to be in the first two 16-bit bytes following the Internet header, source port first and destination port second. The location of this word is found by adding the contents of the Internet data offset field to the address of the zeroth word of the Internet header.

An additional feature of the Internet Queue Mechanism is that it (optionally) provides an logical host capability. That is, incoming messages can be demultiplexed on the basis of bits in the Internet Destination Host field. Exactly which bits are considered logical host specifier bits depends on the naming conventions in force in the network (area) to which the host in question is connected. For ARPANET hosts, the middle 8-bit byte of the 24-bit Internet Host is considered to be the logical host specifier.

ASNIQ takes a specification of what numbers are to be used in the logical host bits of the source address for packets which are sent and received. When a packet is sent, the logical host bits are obtained from the user's packet and combined with the Internet Host Identifier for the host being used, the result being inserted into the Source Host field of the packet. Incoming packets are accepted by the Internet Layer if they match the Internet Host Identifier (Name) when the logical host bits of the destination address have been set to zero.

It is illegal to assign an Internet Queue such that the Logical host mask includes any bits which are not part of the logical host field. ASNIQ will give an error return in this case.

Assign Internet Queue JSYS

```
;T1/    Pointer to queue descriptor block
;       (Bit-0: use Raw Packet Interface)
;T2/    (Unused, must be 0)
;T3/    (Unused, must be 0)
;
;       ASNIQ          (JSYS 756)
;Ret+1:  Failed.  Error code in T1.
;Ret+2: Success.  Internet queue handle in T1.
```

Format of a queue descriptor block:

Word offset	Name	Contents
0	.IQPRV	Internet protocol number in bit 24-31. Others should be 0.
1	.IQFHV	Internet foreign host value word in bits 0-31.
2	.IQSHV	Internet source host value word in bits 0-31. Used for logical host selection.
3	.IQPTV	Internet port value word. Local port value in bits 0-15, foreign port in 16-31.
4	.IQPRM	Mask word corresponding to .IQPRV.
5	.IQFHM	Mask word corresponding to .IQFHV
6	.IQSHM	Mask word corresponding to .IQSHV
7	.IQPTM	Mask word corresponding to .IQPTV (Use 0 for portless protocols)

The mask words specify those bit positions where an exact match is required. Thus, one can make .IQFHM contain 0 in order to talk to all Internet hosts. Or by making say the low 3 bits of the local port mask word be 0, one owns eight ports. Note that an error will result unless the current QDB differs in the masked in bits from all other Internet queues which are assigned at the instant the ASNIQ is issued.

Possible errors:

600737,NTWZX1,NET WIZARD capability required
600740,ASNSX1,All Internet queues in use

600741,ASNSX2,Conflict with some other job (# in AC2)

Release Internet Queue JSYS

```
;T1/     An Internet Queue Handle or -1 for all
;T2/     (Not used.  Must be 0)
;T3/     (Not used.  Must be 0)
;
;        RELIQ   (JSYS 757)
;Ret+1:   Failure.  Error code in T1.
;Ret+2:  Success.
```

This JSYS releases ownership of an Internet queue so that other jobs can assign it. Note that neither the RESET JSYS nor the EXEC RESET command execute RELIQs. Thus, once the queue is assigned, it stays assigned until the job logs out.

Possible errors:

```
600742,SQX1,Internet queue handle out of range
600743,SQX2,Internet queue not assigned
```

Send an Internet Message JSYS

```
;T1/    Internet Queue Handle
;T2/    Pointer to buffer containing message
;T3/    (Not used. Must be 0.)
;
;       SNDIN   (JSYS 754)
;Ret+1:  Failure.  Error code in T1.
;Ret+2: Success.
```

The buffer must contain a word count in the right half of word-0, a valid Internet header in words 1 through 5, and possibly some data in words 6 and following. If port filtering is being used (.IQPTM was non-zero for ASNIQ), the ports must be located in the word following the Internet header. The address of this word is found by adding the address of word-1 in the buffer to the number in the Internet data offset field.

The monitor fills in the source host field in the packet and also the Internet header checksum. The rest of the header is what is supplied by the user.

Possible errors:

```
600742,SQX1,Internet queue handle out of range
600743,SQX2,Internet queue not assigned
600732,SNDIX1,Invalid message size
600733,SNDIX2,Insufficient system resources (No buffers available)
600735,SNDIX4,Invalid header value for this queue
                (Includes Internet Packet Length too big, Data offset
                too small, filtering on ports but packet length says
                packet does not contain a port word, and header does
                not fit the ASNIQ arguments).
```

Receive Internet Message JSYS

```
;T1/    Flags,,Internet Queue Handle
;               Bit-0: On to cause fail return instead of waiting
;T2/    Pointer to buffer where message will be put
;T3/    (Not used.  Must be 0.)
;
;       RCVIN   (JSYS 755)
;Ret+1:  Failure.  Error code in T1.
;Ret+2: Success.
```

Each RCVIN gets one message from the named queue. These messages match the values in the queue descriptor block when masked by the mask words in the block. The number of words filled plus one (counting the count word) is placed in the left half of word-0 of the buffer. If the message was too big as determined by the Internet data length field, as much as will fit in the buffer is transferred and an error return given. No retry for the same message is possible.

Possible errors:

600742,SQX1,Internet queue handle out of range
600743,SQX2,Internet queue not assigned
600732,SNDIX1,Invalid message size

```
; The following structure definitions have been excerpted from
; the file INPAR.MAC which is used in building TENEX and TOPS20
; monitors with Internet code.  Those who are unfamiliar with the
; structure facility should refer to SYS:MACSYM.MAC.  Basically,
; DEFSTR(Name,Location,Bit,Size)  defines a structure called Name.
; This field is held in the word (possibly an index) at Location.
; The right-hand bit number is Bit and the field is Size bits wide.
; A typical reference to the data offset field might be:   LOAD T1,PIDO,(PKT)
; where PKT is an index register which points at the zero-th word
; of the structure.
; Word offsets:

 .IPKVR==0              ; Word with version, type of service, etc
 .IPKSG==1              ; Word with segmentation info
 .IPKPR==2              ; Word with time to live, checksum, protocol
 .IPKSH==3              ; Word with source host
 .IPKDH==4              ; Word with destination host

DEFSTR(PIVER,.IPKVR,3,4)        ; PACKET.INTERNET.VERSION
DEFSTR(PIDO,.IPKVR,7,4)         ; PACKET.INTERNET.DATA-OFFSET
DEFSTR(PITOS,.IPKVR,15,8)       ; PACKET.INTERNET.TYPE-OF-SERVICE
  ; Overlays for the above:
  DEFSTR(PIPRI,.IPKVR,9,2)      ; PACKET.INTERNET.PRIORITY
  DEFSTR(PISVD,.IPKVR,10,1)     ; PACKET.INTERNET.STREAM-VS-DATAGRAM
  DEFSTR(PIREL,.IPKVR,12,2)     ; PACKET.INTERNET.RELIABILITY
  DEFSTR(PISVR,.IPKVR,13,1)     ; PACKET.INTERNET.SPEED-VS-RELIABILITY
  DEFSTR(PISPD,.IPKVR,15,2)     ; PACKET.INTERNET.SPEED-OF-DELIVERY
DEFSTR(PIPL,.IPKVR,31,16)       ; PACKET.INTERNET.PACKET-LENGTH
DEFSTR(PISID,.IPKSG,15,16)      ; PACKET.INTERNET.SEGMENT-ID
DEFSTR(PIFLG,.IPKSG,18,3)       ; PACKET.INTERNET.FLAGS
  ; Overlays for the above:
  ; Unused bit
  DEFSTR(PIDF,.IPKSG,17,1)      ; PACKET.INTERNET.DONT-FRAGMENT
  DEFSTR(PIMF,.IPKSG,18,1)      ; PACKET.INTERNET.MULTIFRAGMENT
DEFSTR(PIFO,.IPKSG,31,13)       ; PACKET.INTERNET.FRAGMENT-OFFSET
DEFSTR(PITTL,.IPKPR,7,8)        ; PACKET.INTERNET.TIME-TO-LIVE
DEFSTR(PIPRO,.IPKPR,15,8)       ; PACKET.INTERNET.PROTOCOL
DEFSTR(PICKS,.IPKPR,31,16)      ; PACKET.INTERNET.HEADER-CHECKSUM
DEFSTR(PISH,.IPKSH,31,32)       ; PACKET.INTERNET.SOURCE-HOST
  ; Overlays for above:
  DEFSTR(PISHN,.IPKSH,7,8)      ; PACKET.INTERNET.SOURCE.NET
  DEFSTR(PISHT,.IPKSH,31,24)    ; PACKET.INTERNET.SOURCE.HOST
DEFSTR(PIDH,.IPKDH,31,32)       ; PACKET.INTERNET.DESTINATION-HOST
  ; Overlays for above:
  DEFSTR(PIDHN,.IPKDH,7,8)      ; PACKET.INTERNET.DESTINATION.NET
  DEFSTR(PIDHT,.IPKDH,31,24)    ; PACKET.INTERNET.DESTINATION.HOST
```