

WARNING - EXPERIMENTERS ONLY!

Host mode on TNC-2 and derivative TNCs at this time remains experimental. Obvious features received a quick "going over" and rather cursory debug, but discovery of the remaining inevitable bugs and critical programming oversights awaits the adventurous reader/developer. Please direct discoveries and criticisms:

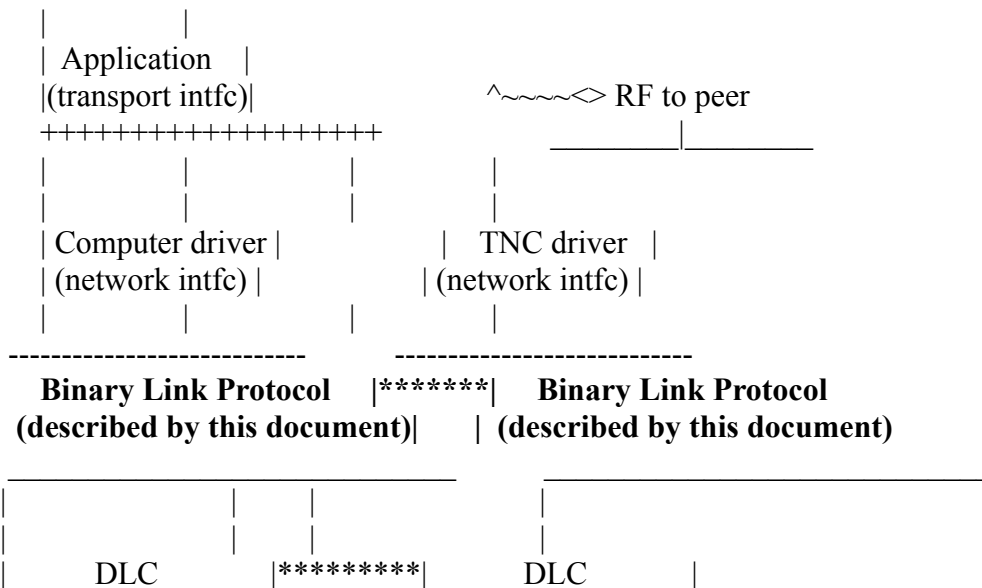
via slow boat : to Howard Goldstein
5201-23rd Ave. N.
St. Petersburg, FL 33710 USA
via compuserv : to 75006,702
via ham packet: to N2WX @ W4DPH.FL

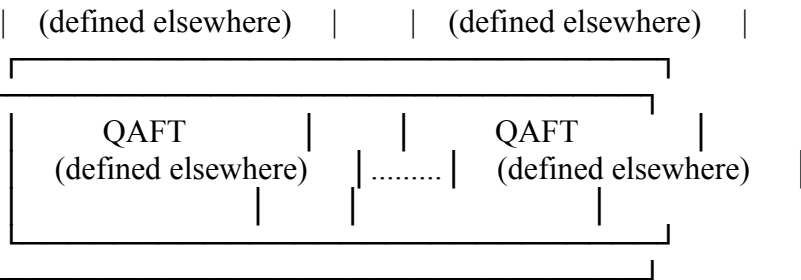
Introduction

This document describes the network layer interface used in TNC-2's binary interface "host" mode protocol, "BLP."

Where it fits in

BLP is a method for interfacing an automated application to a BLP-capable AX.25 terminal node controller.





Key:

- One physical connection
- ***** One or more logical connections

BLP encapsulation

A BLP interface communicates with its peer interface by sending BLP packets to and receiving them from the underlying DLC machine.

BLP packet structure

>| LCNbyte | CTRLbyte | [variable per CTRL] |<

byte :: 8 bit unsigned integer

wd :: 16 bit unsigned integer transmitted as two 'bytes', least significant byte first

LCN : Logical channel to which the BLP packet corresponds.

LCN 0-6F :: Gen'lly, links initiated by a terminal, to a TNC

LCN 70-7f :: Special permanently connected functions

LCN 70 - TNC global command and configuration

LCN 71 - TNC broadcast channel

--- do NOT issue call setup packets to the broadcast channel. TNC-2 may balk if you do.

LCN 72-7f - reserved

LCN 80-EF :: Gen'lly, links initiated by a TNC, to a terminal.

LCN F0-FF :: reserved

CTRL : type of packet

hex

02 CS - Call setup (request)
12 CSEMAIL - Call setup , outgg email ** not TNC-2
04 CCC - Call connected (response)

08 CCLR - Call clear (request)
09 CCLR D - Call cleared (response)

10 CSTENQ - Call status enquiry (request)
11 CSTREP - Call status reply (response)

20 UDATA - Unsequenced data (request)

8X DDATA - Sequenced data (request)

9X DACK - Sequenced data acknowledgement (response)
AX DBUSY - Sequenced data acknowledgement and busy indication (response)

Defining BLP

BLP is a balanced protocol. Since there are no master-slave relationships, any one BLP device can communicate with any other electrically connected BLP device. (for ex., one computer-one TNC, or two computers and no TNCs, or two TNCs back-to-back).

BLP is described and implemented as multiple independent event-driven finite-state machines.

BLP Objects

Each BLP link uses these variables and objects:

BSTATE: One of n states. (see section on BLP states, below)

BQUEUE: FIFO queue of packets as yet unacknowledged by the peer BLP device.

BLP States

There are five Supervisory states and four Data states. The Data states are valid only when the Supervisory state is BSDATA. See the appendix for the state tables.

Supervisory States - C packets

BSIDLE

The BLP link is now inactive and awaits either (1) receipt of a call-setup (*CS*) packet; (2) local command to initiate an outgoing call from the upper-level driver; (3) datagram.

When the state is BSIDLE, the BLP machine is completely idle except that the machine responds to received *CCLR* command packet events with a *CCLRD* indication.

If the BLP machine receives *CS*, i.e. if it is called, it shall respond with a call-accepted (*CA*) packet and start its attempt to set up its upper-level application. For example, if BLP is implemented on a TNC, receipt of a *CS* causes the TNC's AX.25 machine to begin a connection attempt.

BSRCSETUP

The BLP machine enters this state when the local, upper-level application commands one of its idle BLP machines to setup a connection to a remote BLP device. When a BTIMER-expired event occurs, the machine in this state sends a *CS* packet to the peer.

The BLP machine leaves this state when (1) a call accepted (*CA*) indication is received; or (2) a call clear (*CCLR*) command is received; or (3) the retry limit is exceeded; (4) a collision occurs i.e. a *CS* packet is received; or (5) the local application commands the BLP to clear the connection.

BSLCSETUP

A called BLP machine enters this state after it begins initiating a connection with its upper-level driver. If the upper-level driver is successfully connected (ex: TNC able to make its AX.25 connection), the machine enters the BSDATA state. Otherwise, if the connection fails, the machine enters the BSCLEARWT state.

BSCLEARWT

The BLP machine is in this state when a BLP link is being torn down. When a BTIMER-expired event occurs, the machine sends a *CCLR* command. Cleared indication (*CCLRD*) packet received and retry-limit exceeded events cause the machine to enter its idle state.

BSDATA

The BLP machine's local, upper-level application is ready to transfer data with an active remote BLP. All 'D' packets are passed to the Data machine.

DDATA

The remote BLP is ready to receive data packets. The state changes to DWAIT when new outgoing sequenced data packets are enqueued. DDATA state is reentered once all outstanding packets are acknowledged.

DWAIT

The BLP machine has data outstanding for the remote BLP, and is awaiting a response. If a BTIMER-expired event occurs, the data packet is retransmitted. If a DACK or DBUSY packet acknowledging all outstanding packets is received, the state changes to DDATA.

BLP packet types

Supervisory packets

CS - Call setup

format:

[LCN|02|[called_address]][calling_parameters]]

A sequenced BLP link is preceded by the call setup phase. A device directs a CS packet towards another device capable of reaching the [called address].

When a device receives a CS packet, if the LCN is not already in use the device shall bind a local upper level control structure to the LCN, configure the specified calling_parameters, command the upper level to open a link to the called_address specified, and at some time in the future respond with either CCC or CCLRD. For example, a TNC-2 binds the LCN to one of its AX.25 connection-control blocks, starts the connection attempt to the called_address, gets connected, and responds with a CCC.

When a device currently sending CS packets on the LCN (i.e., in the BSRCSETUP state) receives a CS packet, a collision has occurred, and the connection is terminated.

CS address formats:

[called_address]

Null terminated destination AX.25 callsign and digipeaters in AX.25/HDLC format, i.e. left shifted.

[calling_parms]

RFU; use BLP channel to TNC command interface to select these in the meantime.

CCC - Call connected

format:

|LCN|04|

When the device receiving a CS packet has (1) bound the LCN to its upper-level handler, (2) commanded the upper-level to open a connection, and (3) successfully connected the upper-level handler to the network called `_address`, the device sends a CCC and enters the BSDATA state.

CCLR - Call clear

format:

|LCN|08|[reason_code]|

A device attempting to close the LCN indicates the attempt by sending CCLR. The `reason_code` (format to be determined) explains why; ex: upper-level busy, retry failure, call collision, etc.

Meaning of "generic" [reason_code]s:

0:= Remote requested (typical)

1:= BLP couldn't connect LCN (either out of control blocks or failed to find a free channel to connect with or already connected to that station)

Meaning of [reason_code]s only sent by a TNCs with BLP:

2:= Called AX.25 address was busy

3:= (Now disconnected) AX.25 link associated with this address retried out

CCLR D - Call cleared

format:

|LCN|09|

A response to a CCLR command, indicates receipt of the clear command and successful unbinding of the LCN.

CSTENQ - Call status enquiry

format:

|LCN|10|

Causes the device to return a CSTREP status reply packet

CSTREP - Call status reply

format:

|LCN|11|[status]|

Response to CSTENQ enquiry packet. Final format of [status] to be determined; preliminary:

status[0] = LCN's current supervisory state #

status[1] = LCN's current data state #

status[2..n] = LCN's upper level status

ex: TNC responds status[2] = AX.25 state

status[3] = approx # of unacknowledged
AX.25 packets

UDATA - Unnumbered data

format:

|70|20|[called_address]|[data0...datan]|

Broadcast data command packet. Generated by TNC upon receipt of AX.25 UI frame. TNC generates UI frame when it receives a BLP UDATA packet.

Recall that LCN 0x70 is permanently connected; do not attempt to CS channel 0x70 (TNC-2)

DDATA - Sequenced data command packet

format:

|LCN|8x|[data0...datan]|

x :: sequence number, mod 16

DACK - Sequenced data acknowledgement

format:

|LCN|9x|

x :: sequence number, mod 16

Acknowledges packets through sequence number 'x'-1

DBUSY - Sequenced data acknowledgement/busy

format:

|LCN|Ax|

x :: sequence number, mod 16

BLP 3.e Jan 91

Acknowledges packets through sequence number 'x'-1 and inability to receive sequence number x.

SUPERVISORY PACKET RECEIVED EVENTS

	Call Setup	Call connected	Call clear	Call cleared
BSIDLE	Start local, BSstate->BSLCSETUP	Send call cleared, BSstate->BSIDLE		
BSLCSETUP		Stop local, BSstate->BSIDLE		
BSRCSETUP	Stop local, BSstate- (*collision*	BSstate->BSDATA, BDstate->BDIDLE	"	
BSCLEARWT		"		BSstate->BSIDLE
BSDATA		"		

State Changes: Implicit in each described state change is:

1. setting BTIMER to expired, and
2. the zeroing of the Retry count.

Call Status packet: A call-status-reply (*CSTREP*) packet is sent whenever a call-status-enquiry (*CSTENQ*) packet is received.

LOCAL SUPERVISORY EVENTS

	Local outgoing start	Local stop or open failure	Successfully connected to upper, local layer	intentionally blank	Retry-limit exceeded	BTIMER expired
BSIDLE		BSstate->BSRCSETUP				
BSLCSETUP		Stop local, BSstate->BSCLEARWT BDstate->BDIDLE	Send CCC, BSstate->BSCLEARWT	Stop local, BSstate->BSDATA,		BSstate->BSCLEARWT
BSRCSETUP	"		"	Send call setup & restart btimer		
BSCLEARWT	"		BSstate->BSIDLE BTIMER	Send call clear & restart		
BSDATA	"		Stop local, BSstate->BSCLEARWT	(see Data table)		

DATA LEVEL STATE TABLE

Received	New outgoing data are queued	All outgoing data were acknowledged	BTIMER expired, or rxd unsolicited CSTREP	we're busy	Received DDATA but DDATA	Received DACK or DBUSY
BDIDLE			Send DBUSY, DATAPROC ACKPROC			
	BDstate->BDWAIT		BDstate->BDDBSY	(responds with		(signals all-ack'd
		DACK)	events&frees txqueue)			
BDWAIT		Transmit queued	Send DBUSY, " "			
		BDstate->BDIDLE	data, restart	BDstate->BDDBSYWT		
		BTIMER				
BDBSY			"			
		BDstate->BDDBSYWT				
BDDBSYWT		Transmit queued	"			
		BDstate->BDBSY	data, restart			
		BTIMER				
	We're no longer busy	CSTREQ	Received			

