A Packet Broadcast Protocol

J. Gordon Beattie, Jr., N2DSY

The Radio Amateur Telecommunications Society

206 North Vivyen Street
Bergenfield, NJ 07621

Abstract

This paper first identifies the packet radio broadcast functions and operating environment. The apparent simplicity of broadcast protocols can be deceiving. Issues such as the recovery of lost data segments must be addressed in an efficient manner. The must be done with care if the protocol is to be robust and maintain a near-maximum throughput to be practical for file distribution and conferencing. The protocol defined in this paper uses a selective retransmit request capability which allows the users operating without an error to receive subsequent segments while retransmission of the errored segments continues on a round-robin basis. The protocol also provides a control capability (to be defined in a future document) used to establish broadcast file transfers and conferences.

1. Introduction

Packet Radio provides the user community with a mechanism which controls the the flow of relatively great amounts of data between stations. Most bulletin traffic seen by packet users has been forwarded through a series of point-to-point links between Packet Bulletin Board Systems (PBBSs) and then presented to each user on a per request basis. This approach has served the community well and will continue to do so for many years to come. There is a need to improve the functionality of packet mode operations to include support for multi-user file distribution, multi-user nets or conferences, emergency alerting and network management capabilities.

- **File Distribution -** File distribution is quite important to current packet users because the files contain bulletin information used by amateurs active in everything from DX'ing to SKYWARN. Currently, each user logs into the PBBS and retrieves their copy of the bulletins while others wait their turn. With the increased level of activity in most local nets, some users just don't get the information when they need it.

- **Conferences or Nets -** Packet is perceived to be strictly a point-to-point operation lacking the spontaneity of a roundtable or the usefulness of a multi-station directed net. Broadcast operation provides an efficient conversational multi-station capability for packet.

- **Alerts -** Another use for a broadcast-based multi-user packet mode capability is the transmission of alerts or call-outs to ARES or RACES members.

- **Network Management -** The control operators can collect alarms on failed packet links or repeater troubles through packets sent in a broadcast system.

2. The Big Picture

Before defining any broadcast protocol one must understand the operational needs and the operating environment(s) where it will be used. The table below shows three typical operating environments found in packet networks and their associated characteristics. This includes the identification of the technology used to distribute the information to each user, the number of simultaneous users, the link technology used to transport the data, and the recovery mechanism. Based on this information conclusions were made about the relative manageability, overhead and reliability. From the table one can see that there is clear advantage to the use of a network switch as the broadcast point. The network switch assumes it is connected to users with personal computers in their stations, and that they are running the BBC’ software. The BBC package provides the protocol for retransmit requests and other functions needed to administer broadcast distribution of files and the establishment of conferences or nets. Some investigation into an implementation using embedded TNC software is also underway, but no firm plans have yet been established.

Broadcast Distribution Comparison							
Broadcast Point	**Recipient**	**Uplink Type**	**Downlink Type**	**Recovery Type**	**Manageability**	**Overhead**	**Reliability**
"Digipeater'	Many Simultaneously	UI Frames	UI Frames	None	Low	Low	Low
PBBS	One per Connection	LAPB Frames	LAPB Frames	Medium	High	High	Medium
Network Switch*	Many Simultaneously	LAPB Frames BBC	UI Frames BBC	High	High	Low	High

* The definition of a “Network Switch” includes high profile packet switches and satellites.

3. Architecture

The previous sections have defined the user needs and operational environment. In this section we will describe the supported architecture and applications for which the protocol has been designed.

In figure 1 (Broadcast Environment) there is a packet network consisting of five ROSE X.25 Packet Switches. This network has three local networks providing user access and one providing server access. The fifth switch is internal to the backbone and is transparent to all users.

The ROSE X.25 Switch serving the ROSErvers is shown sending a network alarm contained in a level 2 Unnumbered Information (UI) frame to one of the ROSErvers. This is an unacknowledged message and provides the system with the information needed to track down problems when the network manager is available. The alarm type and parameters may be remotely added or deleted as shown in-the top and bottom of figure 2.

Three of the ROSE X.25 Switches have the BBC Broadcast application loaded in the switch. The two ROSE X.25 Switches in “Network Broadcast Operation” are receiving their broadcast data in real-time from the ROSErver, and are sending it out to their users in level 2 Unnumbered Information (UI) frames. To set this up, the ROSErver made a X.25 level 3 connection to the BBC Broadcast application in each switch, then sent control data to the ROSE X.25 Switch BBC application. This

[1] The “BBC” is a set of application software which runs on MS-DOS™ computers and ROSE X.25 Switches. The author felt that no name was better known in the broadcast industry than the “BBC”, and so named the system “BBC”. It should be noted however that there is no relationship to the British Broadcasting Corporation, leaving open the possibility that the author had other motives for naming the system “BBC”.

control information, included the distribution group, the time of transmission (now), recovery timers and counters required for the broadcast. If the file is short, the ROSErver can load the control information, and the broadcast file into the ROSE X.25 Switch and then disconnect. The file broadcast would occur at the time specified without further participation from the ROSErver. The setup aspects of the broadcast may be set by predefined controls or through the mechanisms shown in figure 3. Figure 4 shows the broadcast file distribution flow. The control protocol data units shown in the figures are not defined in this paper[2] and are optional at this time. See footnote 2.

In the local network at the bottom of the figure, the group there has a net or conference in progress which allows any user to send data over a standard LAPB AX.25 link to the ROSE X.25 Switch BBC -application. The BBC application then broadcasts the data using UI frames to the whole group. Recovery is provided for any data that is lost. See figure 6 for the event flow. The control protocol data units shown in the figures are not defined in this paper and are optional at this time. See footnote 2.

4. Broadcast Protocol Data Units

The Broadcast Protocol Data Units (BPDUs) listed below provide structure to the messages specific to particular applications. The five BPDU types are:

Broadcast PDU Types	
BPDU Type	Use
Event Report	Unconfirmed data unit User or File Data
Create Request	Confirmed data unit Control and User Stations
Create Response	Confirmation of Create Request
Delete Request	Confirmed data unit Control and User Stations
Get Request	Confirmed data unit Control and User Stations
Get Response	Confirmation of Get Request
Set Request	Unconfirmed data unit Control Stations

The Event Report PDU provides the user with an envelope for user or file data in the broadcast application. No specific acknowledgement is required for this data when transmitted in a UI frame, but it may be provided by underlying protocols such as **AX.25** or X.25.

The Create Request and Response PDUs provide the control station with the capability to establish broadcast Master Control Blocks which contain parameters needed to control file transfers and conferences, or nets.

The Delete Request and Response PDUs provide the control station with the capability to remove broadcast Master Control Blocks.

[2] The control protocol data units (CREATE, DELETE, GET & SET) and associated data structures are to be defined in a later paper.

The Get and Set PDUs allow a control station to read and modify control parameters contained in a broadcast Master Control Block.

5. Event Report Messages

The basic protocol data unit (PDU) in the broadcast application and in the BBC software is the Event Report. It is the basis for several message types which contain the broadcast and recovery data. Event Reports may be sent via UI frames or via connected mode packet protocols such as AX.25 and X.25. There are several message types based on this PDU. They all use the structure of the Event Report PDU to convey data specific to the application. This is the only required PDU. The message types based on this PDU are:

Event Report Message Types	
Message Type	Description
File Header	Provides Context Information
Data	Provides File Data
Request Transmission	Requests Transmission of Files or Missing Segment(s)
End Of File	Signals End of Transmission

The notation used to define data types and lengths in this document is listed in the tables below.

Int-1	one byte unsigned integer	range 0 - 255
Int-2	two byte unsigned integer	range 0 - 65,535
Int-3	three byte unsigned integer	range 0 - 16,777,215

Integers are sent low order byte first. Printable Strings are sent left most byte first.

5.1 File Header Message

The File Header message type provides the context information needed to handle the file in the receiving user's system. This header will be preserved by the broadcasting device for the duration of the broadcast session. This message is sent when a Request Transmission message is received with a segment number of 0.

File Header Message			
Element	**Value**	**Data Type**	**Notes**
Version	0	Int- 1	Current Version
Event Report	0	Int- 1	PDU Type
File Header	0	Int- 1	Message Type
Originator	Callsign	PS-10	Left justified, pad with 20H
BroadcastTitle	Purpose	PS-30	Left justified, pad with 20H
BroadcastName	Filename	PS-20	Left justified, pad with 20H
BroadcastType	Content Encoding	PS-10	Left justified, pad with 20H
BroadcastAlias	Session Identifier	Int-4	range limited 0- 16,777,2 15
BroadcastState	Enumerated	Int-1	See Values below
SegmentSize	Integer	Int-2	Needed for later recovery
SegmentCount	Integer	Int-4	Number of blocks

Field Definitions

- **Version -** The version number is defined for this release to have the value 0.
- **Event Report -** This field is defined to be inform the receiving system that this message is an event report PDU.
- **File Header -** This field is defined to inform the receiving system that this message is a file header.
- **Originator -** Callsign of the information source.
- **Broadcast Title -** Activity Name. Some consideration is being given to registering these to ensure uniqueness.
- **BroadcastName -** This is the filename of the file being sent.
- **BroadcastType -** Content encoding is sent in this field. Standard values are:

ASCII	Text and the control characters defined as ASCII
BINARY	Generic eight-bit data
VOICE	Generic voice traffic
VIDEO-CCC	The string "VIDEO-" followed by the IS0 3166 Alpha-3 Code related to the video standard in use.

- **BroadcastAlias -** This is the short tag sent with each data segment event report in a given broadcast session. It is more compact than the Originator, BroadcastTitle, and BroadcastName. It should not be reused by a given station for long periods of less than 30 days.
- **BroadcastState -** This parameter informs the user of the current distribution and recovery state of the file.

(0) PENDING	- File loaded but not sending
(1) ACTIVE	- File loaded and file being sent
(2) HEADER	- File is being sent but only the header was stored.

- **SegmentSize -** This field provides the number of bytes in a segment. This value provides the blocking information needed to retrieve a segment. This will be quite useful when requesting fills from PBBS-type servers. These servers will often set up broadcasts with a variety of segment sizes depending on the environment. The default value is 240.
- **SegmentCount -** This field provides the total number of segments in the broadcast.

5.2 Data

The Data message type provides the information content needed to transport and correlate elements of the file in the receiving user's system. This message is sent in sequence by the broadcaster and in a round robin fashion when a Request Transmission message is received with a matching segment number.

Data Message			
Element	**Value**	**Data Type**	**Notes**
Version	0	Int- 1	Current Version
Event Report	0	Int-1	PDUType
Data	1	Int-1	Message Type
BroadcastAlias	Session Identifier	Int-4	range limited O-16,777,216
SegmentSize	Integer	Int-2	bytes in segment
SegmentNumber	Integer	Int-4	Number of this segment
SegmentData	User Data	bytes	SegmentSize number of bytes

Field Definitions

- **Data -** This field is defined to be inform the receiving system that this message is a data message.
- **SegmentSize -** This field will match the value given in the File Header when broadcasting files. When broadcasting conference or net messages this value may be ignored.
- **SegmentNumber -** This field provides the sequential sequence number for this segment.
- **SegmentData -** This field will default to 240 bytes of user data when the packet size is defined to be 256 octets.

5.3 Request Transmission Message

The Request Transmission message type provides the user with the capability to request retransmissions of missing segments, or the transmission of a file header or file header and file. This message is sent at any time by the user and causes transmissions or retransmissions to be made in a round robin fashion.

Request Transmission Message			
Element	**Value**	**Data Type**	**Notes**
Version	0	Int- 1	Current Version
Event Report	0	Int-1	PDUType
Request Transmission	2	Int-1	Message Type
BroadcastAlias	Session Identifier	Int-4	range limited 0-16,777,216
Segment&e	Integer	Int-2	bytes in segment
BroadcastRequestType	Enumerated	Int-1	See values below
SegmentA	Integer	Int-4	Recovery point
SegmentB	Integer	Int-4	Recovery point

Field Definitions

- **Request Transmission -** This field is defined to inform the receiving system that this message is a Request Transmission message.
- **SegmentSize -** This parameter conveys the segment size originally specified in the File Header. This field will be most useful for "after broadcast" queries to a fileserver. It provides the server with a stepping index which will match the SegmentSize of the original broadcast. The value of this field

may be ignored by the receiving system when broadcasting conference or net messages.

- **BroadcastRequestType -** This field tells the broadcaster what segments to send the user. Standard values are listed below.

0	>	Greater than the segment number specified in SegmentA. SegmentB is not used.
1	<	Less than the segment number specified in SegmentA. SegmentB is not used.
2	-	All between the SegmentA value and the SegmentB value.

- **SegmentA -** This is the segment number used with the BroadcastRequestType field.
- **SegmentB -** This is the high order segment number value field used when the BroadcastRequestType field is set to 2.

5.4 End Of File Message

The End Of File (EOF) message type provides the user with an indication that this message is the last segment in the file. When broadcasting conference or net messages, this message serves as an indication that the operation is closed.

End Of File Message

Element	**Value**	**Data Type**	**Notes**
Version	0	Int- 1	Current Version
Event Report	0	Int- 1	PDU Type
EOF	3	Int- 1	Message Type
BroadcastAlias	Session Identifier	Int-4	range limited 0-16,777,216
SegmentSize	Integer	Int-2	bytes in segment
SegmentNumber	Integer	Int-4	Number of this segment
SegmentData	User Data	bytes	SegmentSize number of bytes

Field Definitions

- **End Of File -** This field is defined to inform the receiving system that this message is an End Of File message.
- **SegmentSize -** This value is the only time in a file transfer where the value can be different than the one given in the file header.

5.5 Conference Header Message

The Conference Header message type provides the context information needed to set up a conference for users. A conference is a similar to a roundtable type net. It operates in any way the organizer and participants may wish. Dialogue control is not regulated by the broadcaster but by the participants. This header will be preserved by the broadcasting device for the duration of the broadcast session. This message is sent when a Request Transmission message is received with a segment number of 0.

The Conference Header message is structured like a File Header message but there are some changes designed to simplify human operation. Once established, the Conference users will use the same messages as is used for broadcast file operation.

Conference Header Message			
Element	**Value**	**Data Type**	**Notes**
Version	0	Int- 1	Current Version
Event Report	0	Int-1	PDU Type
Conference Header	4	Int-1	Message Type
Originator	Callsign	PS-10	Left justified, pad with 20H
BroadcastTitle	Purpose	PS-30	Left justified, pad with 20H
BroadcastName	Session Name	PS-20	Left justified, pad with 20H
BroadcastType	Content Encoding	PS-10	Left justified, pad with 20H
BroadcastAlias	Session Identifier	PS-4	four printable characters
Broadcasts tate	Enumerated	Int-1	See Values below
Segment&e	Integer	Int-2	Needed for later recovery
SegmentCount	Integer	Int-4	Number of blocks

Field Definitions

- **Version -** The version number is defined for this release to have the value 0.
- **Event Report -** This field is defined to be inform the receiving system that this message is an event report PDU.
- **Conference Header -** This field is defined to inform the receiving system that this message is a conference header.
- **Originator -** Callsign of the user.
- **Broadcast Title -** Conference or Net Name. Some consideration is being given to registering these to ensure uniqueness.
- **BroadcastName -** This is the Session name for the session.
- **BroadcastType -** Content encoding is sent in this field. Standard values are:

ASCII	Text and the control characters defined **as ASCII**
BINARY	Generic eight-bit data
VOICE	Generic voice traffic
VIDEO-CCC	The string “VIDEO-” followed by the IS0 3 166 Alpha-3 Code related to the video standard in use.

- **BroadcastAlias -** This is the short 4 character tag sent with each data segment event report in a given conference session. It is more compact than the Originator, BroadcastTitle, and BroadcastName. It may be re-used as often as is deemed desirable by those managing the conference or net.
- **BroadcastState -** This parameter informs the user of the current distribution and recovery state of the conference or net.

(0)	PENDING	- Conference loaded but not sending
(1)	ACTIVE	- Conference loaded and data being sent
(2)	HEADER	- Conference data is being sent

but only the header was stored.

- **SegmentSize -** This field provides the number of bytes in a segment. This value provides the blocking information needed to retrieve a segment. This will be quite useful when requesting fills from PBBS-type servers. These servers will often set up broadcasts with a variety of segment sizes depending on the environment. The default value is **240.**

- **SegmentCount -** This field provides the total number of segments in the broadcast.

6.0 TNC Settings

TNC settings for broadcast operation are divided into two basic groups: user and broadcaster. The user needs to set up the TNC as follows:

MON	**ON**
MCON	**ON**
MCOM	**OFF**
BUDLIST	**ON**
LCALLS	**callsign** of broadcaster
USERS	1
PACL	0 (256)
SENDPAC	0
CR	OFF
CPAC	ON
PACTIME	**AFTER** 60
DWAIT	20
TXD	35
FRACK	15

The broadcaster needs to set up the TNC as follows:

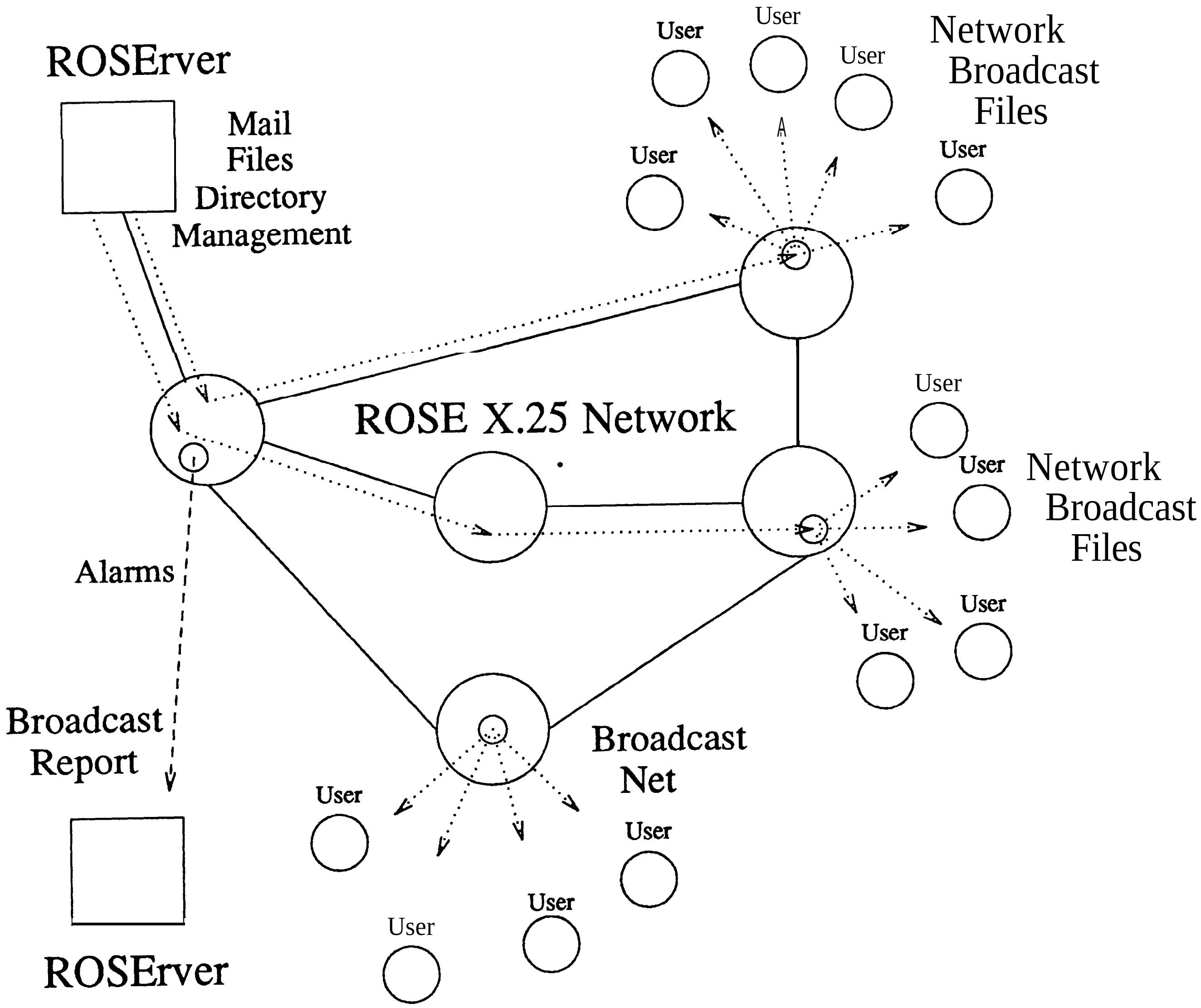
MON	**OFF**
MCON	**ON**
MCOM	**OFF**
USERS	One less than the maximum value
PACL	0 (256)
SENDPAC	0
CR	OFF
CPAC	ON
PACTIME	**AFTER** 10
DWAIT	3
TXD	100
FRACK	3

Additional parameters might be set for transparent binary operation. **These** vary with make and model. See the manual for details.

7.0 Conclusion

This paper has described the broadcast environment, the user applications and the protocols needed to make it all work. The Event Report portion of the broadcast protocol specification is outlined for those who might wish to experiment with this interesting area. Additional broadcast message types and the broadcast control protocol will be added to the specification and published in Amateur circles in the near future. Comments and suggestions for additions and changes would be welcome and are requested. If we can be of assistance with your experimentation, contact us by mail or phone.

Broadcast Environment



J. Gordon Beattie, Jr. N2DSY
28 August 1989

Broadcast Report - Event Flow

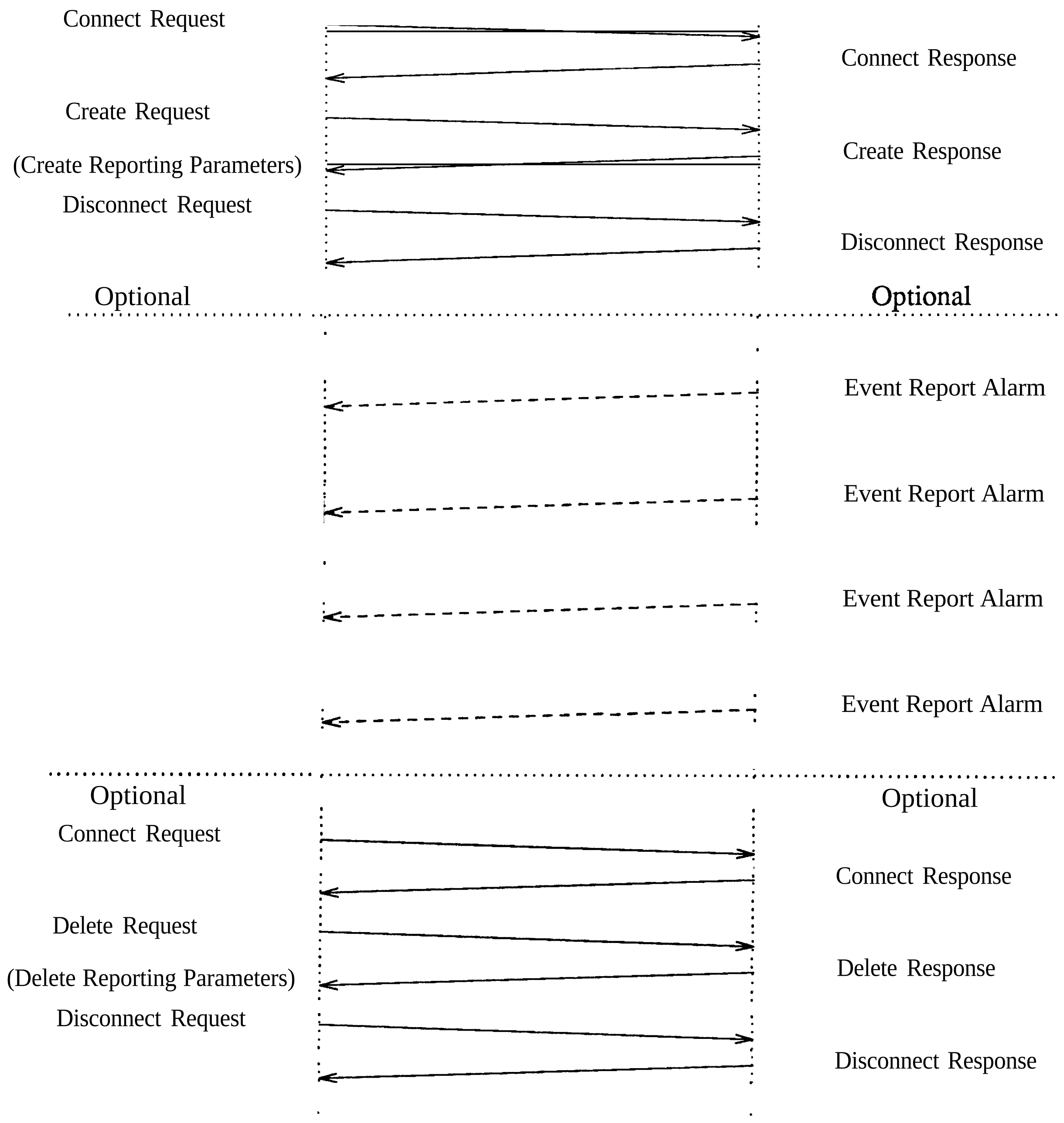


Figure 2

J. Gordon **Beattie,** Jr. **N2DSY**
28 August 1989

Broadcast Setup Flow

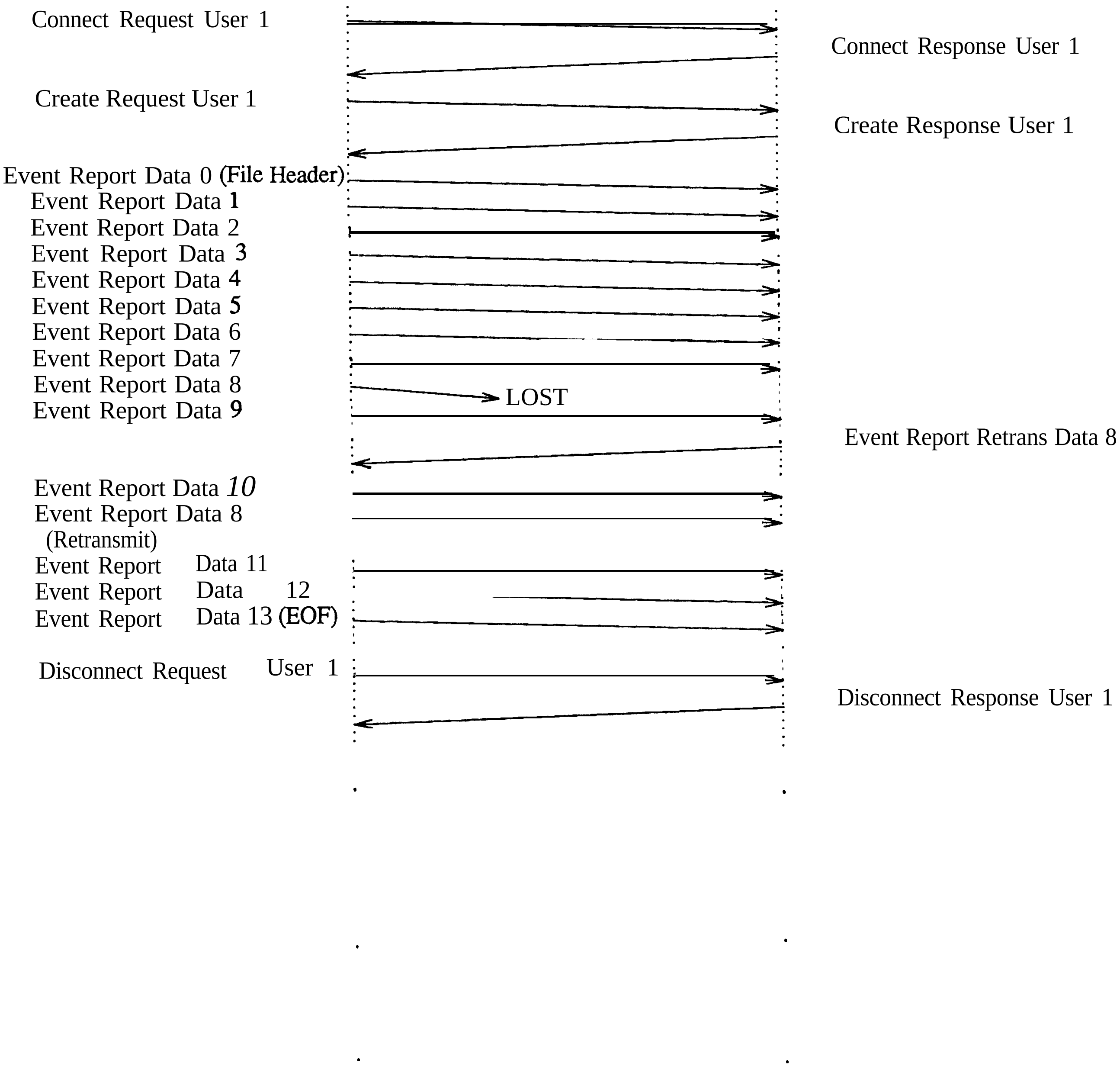


Figure 3

J. Gordon Beattie, Jr. N2DSY
28 August 1989

Broadcast Flow - File Operations

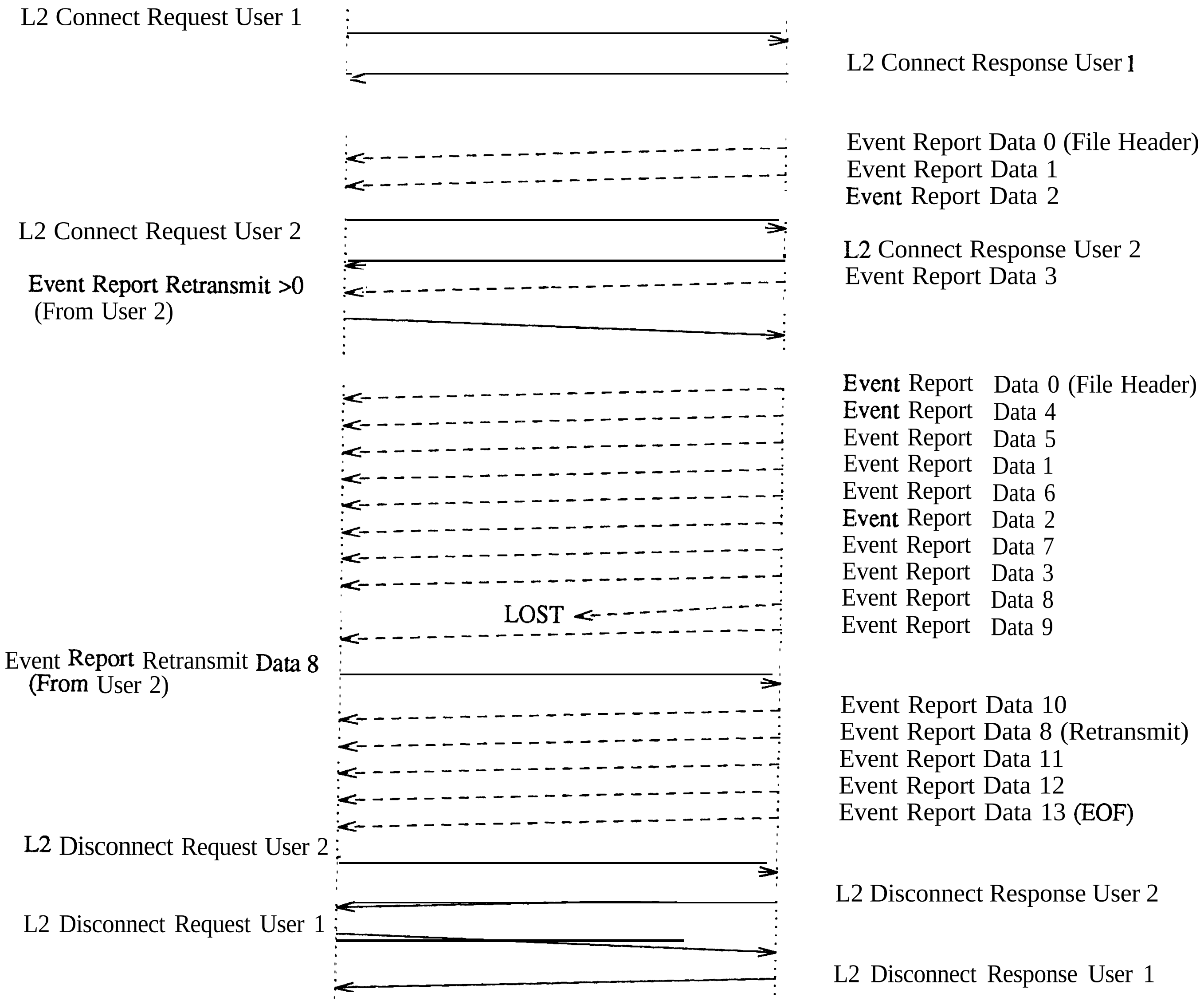

L2 Connect Request User 1
L2 Connect Response User 1
Event Report Data 0 (File Header)
Event Report Data 1
Event Report Data 2
L2 Connect Request User 2
L2 Connect Response User 2
Event Report Data 3
Event Report Retransmit >0
(From User 2)
Event Report Data 0 (File Header)
Event Report Data 4
Event Report Data 5
Event Report Data 1
Event Report Data 6
Event Report Data 2
Event Report Data 7
Event Report Data 3
LOST
Event Report Data 8
Event Report Data 9
Event Report Retransmit Data 8
(From User 2)
Event Report Data 10
Event Report Data 8 (Retransmit)
Event Report Data 11
Event Report Data 12
Event Report Data 13 (EOF)
L2 Disconnect Request User 2
L2 Disconnect Response User 2
L2 Disconnect Request User 1
L2 Disconnect Response User 1

Figure 4

J. Gordon Beattie, Jr. N2DSY
28 August 1989

Broadcast Change Flow

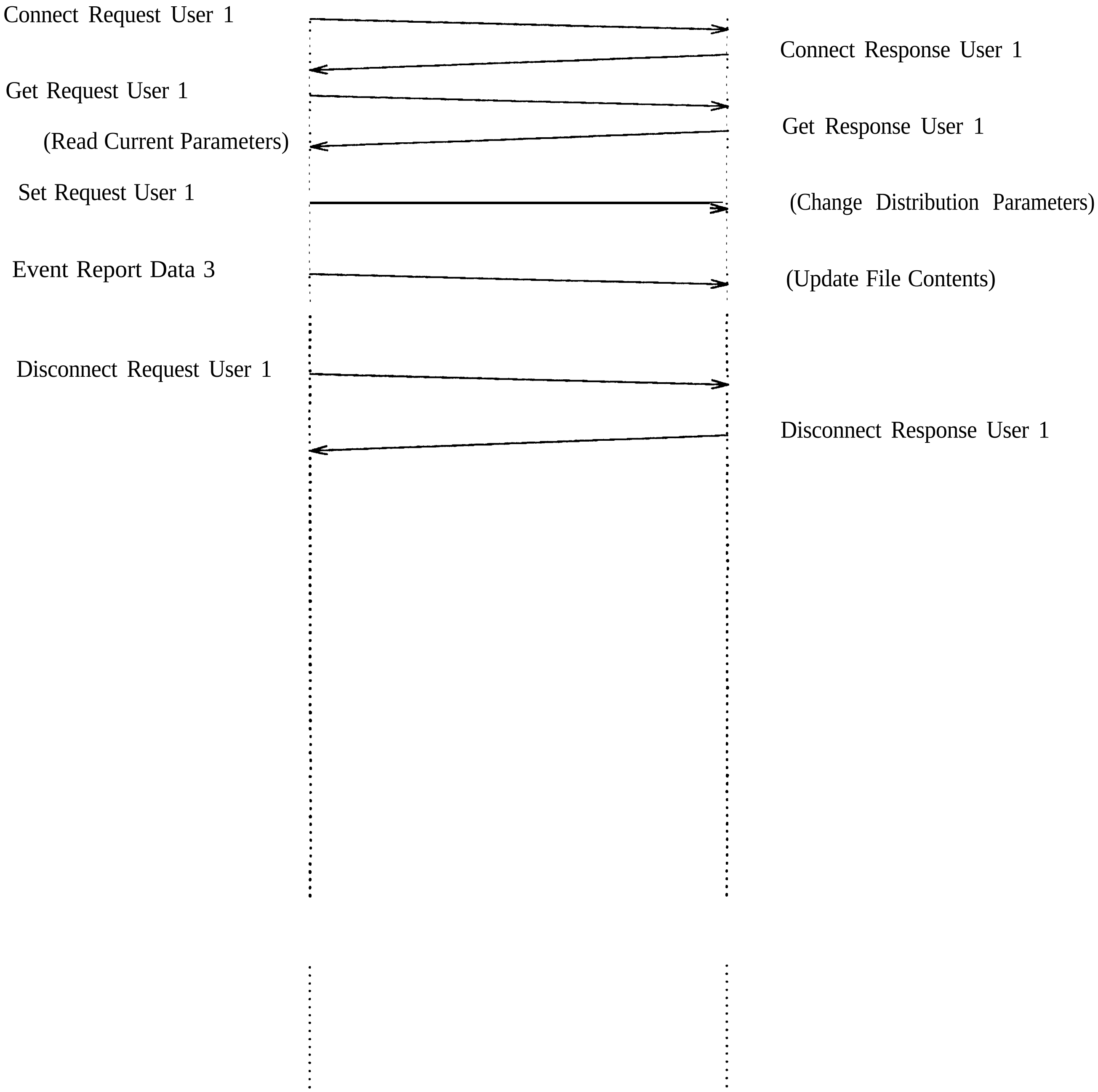


Figure 5

J. Gordon **Beattie,** Jr. **N2DSY**
28 August 1989

Broadcast Flow - Net Operations

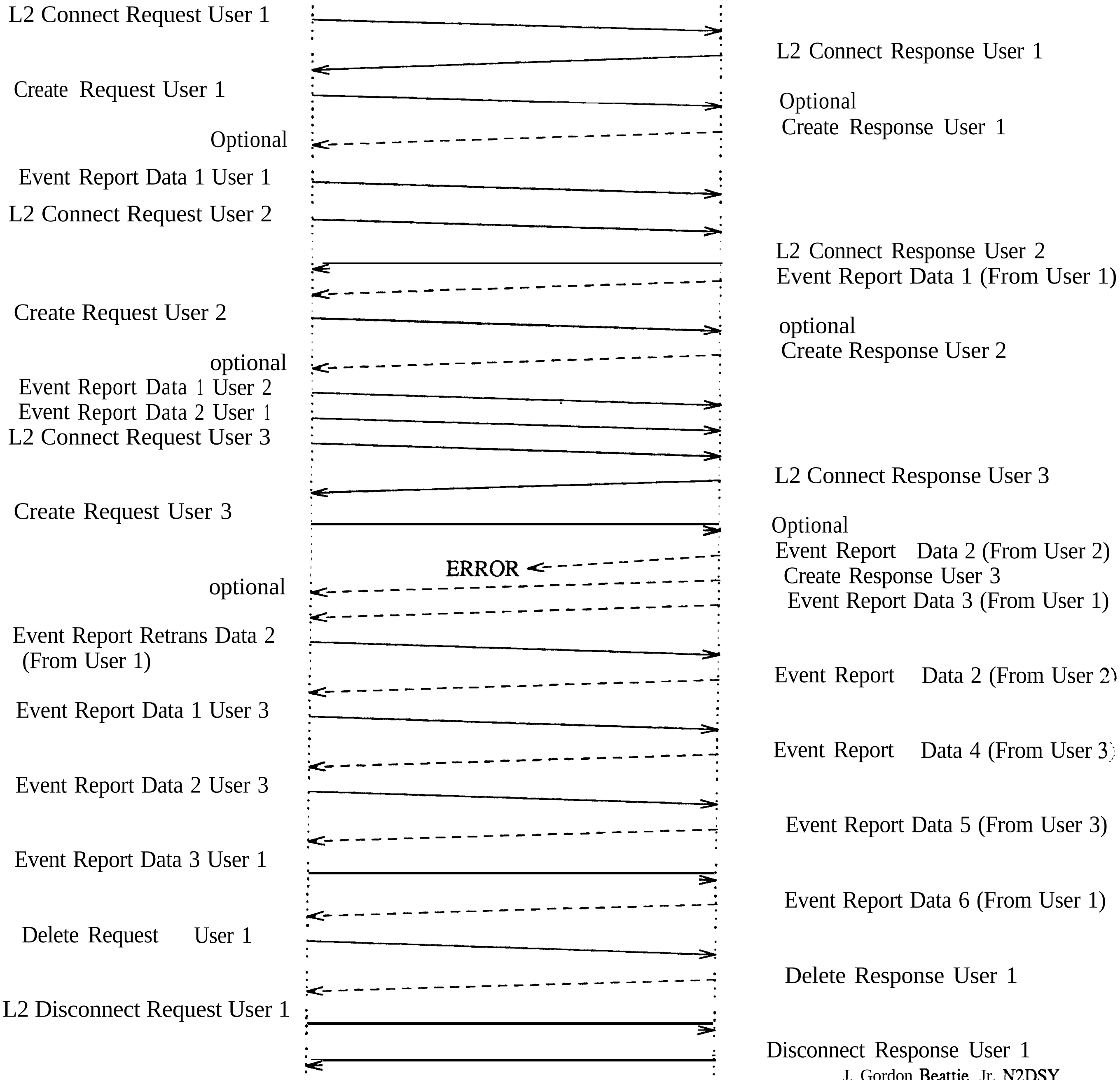

L2 Connect Request User 1
L2 Connect Response User 1
Create Request User 1
Optional
Create Response User 1
Optional
Event Report Data 1 User 1
L2 Connect Request User 2
L2 Connect Response User 2
Event Report Data 1 (From User 1)
Create Request User 2
optional
Create Response User 2
optional
Event Report Data 1 User 2
Event Report Data 2 User 1
L2 Connect Request User 3
L2 Connect Response User 3
Create Request User 3
Optional
Event Report Data 2 (From User 2)
ERROR
Create Response User 3
optional
Event Report Data 3 (From User 1)
Event Report Retrans Data 2
(From User 1)
Event Report Data 2 (From User 2)
Event Report Data 1 User 3
Event Report Data 4 (From User 3)
Event Report Data 2 User 3
Event Report Data 5 (From User 3)
Event Report Data 3 User 1
Event Report Data 6 (From User 1)
Delete Request User 1
Delete Response User 1
L2 Disconnect Request User 1
Disconnect Response User 1

Figure 6

J. Gordon Beattie, Jr. N2DSY
28 August 1989