Network Working Group
Request for Comments: 31

BINARY MESSAGE FORMS IN COMPUTER NETWORKS

Daniel Bobrow
Bolt, Beranek, and Newman
Cambridge, Massachusetts

William R. Sutherland
MIT Lincoln Laboratory
Lexington, Massachusetts

February 1968

MESSAGE FORMS IN COMPUTER NETWORKS

INTRODUCTION

Network communication between computers is becoming increasingly important. However, the variety of installations working in the area probably precludes standardization of the content and form of inter-computer messages. There is some hope, however, that a standard way of defining and describing message forms can be developed and used to facilitate communication between computers. Just as ALGOL serves as a standard vehicle for describing numerous algorithms, and BNF serves as a standard for describing language syntax, a message description language would be useful as a standard vehicle for defining message formats.

Considerable progress has been made at the low level of message handling protocol and one can expect the ASCII protocols to be used. The discussion which follows assumes that the mechanics of exchanging messages, check sums, repeat requests, etc., have been worked out. The topic of concern is how to describe the content and intent of a binary message body when the network header and trailer details have been stripped off.

Most attempts at describing the content of binary messages jump immediately into a consideration of the bit codings to be used. Long, thin rectangles are drawn to represent the binary bit stream; this stream is sliced up into boxes, and tables generally describe the bit options for each box. A better approach would be to provide a symbolic method for describing messages. The symbolism, by avoiding immediate references to specific bit details, should help one’s understanding of the message content and the alternatives available in the message body. When the basic form of the binary message body is clear, the coding details of the actual bit fields can be shown.

Describing a binary message body is not much different from describing a text body or language. Text assumes fixed bit fields each containing one character. Standard language description methods (BNF) then show how the characters can be concatenated and what interpretation should be placed on character groups. Binary message descriptions require the additional capacity of defining various size fields in the message and the interpretation to be placed on the bits contained in the field.

A message description is initially intended as a reference standard to be written down on paper and made available to new users of a computer network. From this standard, the new user can discover the kind and form of the binary data being exchanged over the network. Once this is known, the programs necessary for using the network facilities can be created. Later on, in an established network, one can envision the promulgation of standards for newly developed binary formats via the exchange of ASCII text messages over the network itself instead of on paper through the mail. Still farther into the future, the text of a binary format standard could be used as input to compiler-like programs which automatically create data translation programs for converting one binary format to another. Right now, though, some kind of binary data description method, however trivial, is desperately needed.

A SUGGESTED BINARY FORMAT DESCRIPTION METHOD

The basic component of a binary message is a simple field consisting of a consecutive number of bits in the message. Binary messages consist of concatenated fields. A format description for a binary message will consist of a title and four declarative sections.

1) Symbolic names are declared for all the different kinds of fields found in the binary format being defined.
2) Symbolic names are declared for commonly used values of particular fields.
3) The legal ways of concatenating fields are indicated.
4) The number of bits in each field and any special considerations of bit codings are declared.

The following is a complete example of a binary message description for a trivial kind of pictorial data.

Title: Illustrative graphic data format for a hierarchally structured picture of lines and points.
Simple Fields:
OPT - Option Control Field
COORD - Numerical Coordinate Value
ID - Identnumber for group of picture parts
COUNT - Number of units in message

Field Equivalents:
PHDR <- '2' OPT
LHDR <- '4' OPT
GRPHDR <- '1' OPT
GRPEND <- '3' OPT

```
Characterizations:
   CPAIR   <- COORD = 2
   POINT   <- PHDR + CPAIR
   LINE    <- LHDR + CPAIR = 2
   PARTS   <- POINT/LINE/PARTS + PARTS
   PIXUNIT <- GRPHDR + ID + PARTS + GRPEND
   PIXMSG  <- '5' OPT + N: COUNT + PIXUNIT = N + '0' OPT
Simple Field Sizes:
   OPT   3
   COORD 14
   ID    9
   COUNT 6
```

Declaration of Simple Fields

The declaration of a simple field includes a symbolic name, and for lack of a better way, an English description of what the contents of the field represent. For example:

```
Simple Fields:
   F1    - Geometric Options
   EXP   - STD Number - Exponent
   COORD - STD Number - Geometric Coordinates
```

Representing Field Values

A field with a specific value can be represented by a number in single quotes followed by the field name. A number consists of standard digits construed as binary if zeros and ones. Other numbers must be followed by a base indicator unless no confusion is possible; Q is octal, D is decimal.

```
Example:
'1001' F1
'300D' COORD
'27Q'   EXP
```

Field values are integer numbers assigned such that the least significant bit is sent first. Only that part of the number which fits the field is used. Appropriate sign extension is needed for negative numbers and for numbers whose bit representation is smaller than the field.

Simple Field Equivalents

The declaration of a Simple Field Equivalent provides a symbolic name which represents a particular field with a specific value. Example:

Field Equivalents:
C1 <- ’1001’ F1
C2 <- ’1010’ F1

Characterization Statement

A characterization statement defines a complex field (message or message part) by indicating how other fields can be combined and is similar to a definition statement in BNF. The left side is a complex field name separated (by <-) from the concatenation indications on the right. Field names or equivalent names are concatenated by plus (+), alternatives indicated by slash (/). Slash has precedence over plus so that A + B/C means A followed by either B or C. Alternatives must be distinguishable in their own right.

Characterization statement parts can be grouped in the normal manner by parentheses. (A + B)/C means either A followed by B or C.

Repetition Indicators

Repeated occurrences of a field may be indicated by following the field name with an equal sign (=) and a number. For example:
CPAIR <- (COORD = 2) i.e. exactly two COORD fields
PPAIRS <- (C1 + CPAIR = 10D) / (C2 + CPAIR = 40D)

Assignments Within a Characterization Statement

Simple fields interpretable as integers can be assigned to a variable within the right side of a characterization statement. This variable can then be used as a repetition indicator. Example:

MS <- N1: EXP + CPAIR = N1

indicates that MS consists of field EXP interpreted as an integer and then exactly that number of CPAIRS. All variables are global in scope.

Conditional Fields

Within a characterization statement a field may or may not occur depending on the contents of some other previous field. This situation is indicated by assigning a label to the determining field. The conditional occurrence is then indicated by enclosing a condition expression and the optional field description in brackets ([and]). For example:

SS <- V:F1 + CPAIR + [V = C1 > PPAIRS]
which defines a format of 2 and perhaps 3 fields.
a) Field F1 labeled V followed by
b) Field CPAIR followed by
c) Field PPAIRS if the first field (V) was C1; otherwise, this third field is not present in the message.

Conditional Alternatives

Alternatives selected by the contents of some previous field rather than by the contents of the alternative field itself are indicated by an extension of the conditional field notation. For example:

SM := W : F1 + CPAIR + [W = C1 > CPAIR / C2 > PPAIRS /

The determining field occurs at the beginning of the conditional alternative and each alternative then includes its value for the determining field and the alternative field then present.

Size of Simple Fields

A separate field size declaration is provided.

Simple Field Sizes:

```
F1    4
EXP   7
COORD 12
```

This size declaration should appear at the end of the message description; thus, forcing the reader to postpone an early consideration of bit details. xmodmap -e "add lock = Caps_Lock"