

Submitted for recognition as an American National Standard

Standards for ATIS Message Sets Delivered over Reduced Bandwidth Media

Foreword—This document outlines the US standard for sending Advanced Traveler Information Services (ATIS) messages over reduced bandwidth media data streams. It specifies the “over the air” format and method for such data messages at a bit level within the structure of the host system. The methodology allows a cooperative sharing of the media’s total bandwidth with other non-ATIS data services. It allows for further expansion to other ATIS messages such as transit schedules and weather use. It uses a common directory of messages to divide out those messages which are the subject of this specification from others. It can be deployed upon multiple physical modulation types, typically wireless in nature.

TABLE OF CONTENTS

1.	Scope	6
2.	References	6
2.	Related Documents	6
3.	Definitions.....	7
4.	Overview	12
4.1	Message Types Supported	12
4.2	The Protocol Stack	15
4.3	Physical Modulation	15
4.4	The Network Layer	15
4.5	Data Tokens and Stream	16
4.6	The Examples Used in this Document	17
5.	Background Concepts	17
5.1	Stream Schemas – How the information streams work.....	17
5.2	Tokens and Tables – The Theory of Data Compression Used	21
5.3	Tiles and Grids – Efficient Representations of Latitude and Longitude Values.....	26
5.4	Traffic Models – Shared Models of Link Impedance Modeling.....	30
5.5	Forward Pointers – Indirection to Other Data Sources.....	36
5.6	Frame Bit Stuffing – Unused “Flea Bits” in the Message Frame	37
5.7	Frame Headers – The Concept of Context.....	37
5.8	Transit Scheduling – Unified Concepts of Multi Modal Scheduling Tables	37
6.	Network/Directory Messages.....	38

SAE Technical Standards Board Rules provide that: “This report is published by SAE to advance the state of technical and engineering sciences. The use of this report is entirely voluntary, and its applicability and suitability for any particular use, including any patent infringement arising therefrom, is the sole responsibility of the user.”

SAE reviews each technical report at least every five years at which time it may be reaffirmed, revised, or cancelled. SAE invites your written comments and suggestions.

TO PLACE A DOCUMENT ORDER; (724) 776-4970 FAX: (724) 776-0790
SAE WEB ADDRESS <http://www.sae.org>

Copyright 2000 Society of Automotive Engineers, Inc.
All rights reserved.

6.1	Overview of Structure & Stream Mapping	38
6.2	Message and Table Stream Numbering	38
7.	Message Frame Headers Elements (Context).....	38
7.1	Context Definition for: Binary Data	41
7.2	Context Definition for: Expansion Tables (General Table Use)	42
7.3	Context Definition for: Forward Referencing Tables	44
7.4	Context Definition for: Grid Array Tables	45
7.5	Context Definition for: Incident Flow Data	46
7.6	Context Definition for: Incident Modeling Tables	51
7.7	Context Definition for: Incident Numbering Tables	52
7.8	Context Definition for: Map Encoding (General Base Map Use).....	53
7.9	Context Definition for: Private Data	55
7.10	Context Definition for: RDS Incident Encoding Table	56
7.11	Context Definition for: Resource Availability (Parking Lot Availability, Ride Share)	56
7.12	Context Definition for: Textual Incident Encoding Table	57
7.13	Context Definition for: Transits Schedule	57
7.14	Context Definition for: Weather Data	58
7.15	Context Definition for: Vanity Maps	58
7.16	Context Definition for: Alternative Names Tables	59
7.17	Context Definition for: Table of Tables	59
8.	Message Specifications	60
8.1	Message Definition for: Binary Data	61
8.2	Message Definition for: Expansion Tables (General Table Use).....	61
8.3	Message Definition for: Forward Referencing Tables.....	63
8.4	Message Definition for: Grid Array Tables	64
8.5	Message Definition for: Incident Flow Data	64
8.6	Message Definition for: Incident Modeling Tables	68
8.7	Message Definition for: Incident Numbering Tables	69
8.8	Message Definition for: Map Encoding (General Base Map Use)	69
8.9	Message Definition for: Private Data	72
8.10	Message Definition for: RDS Incident Encoding Table	72
8.11	Message Definition for: Resource Availability (Parking Lot Availability, Ride Share)	73
8.12	Message Definition for: Textual Incident Encoding Table	73
8.13	Message Definition for: Transits Schedule	73
8.14	Message Definition for: Weather Data	74
8.15	Message Definition for: Vanity Maps	74
8.16	Message Definition for: Alternative Names Tables	74
8.17	Message Definition for: Table of Tables	74
9.	Data Elements (DEs)	75
9.1	Name Fields.....	75
9.2	Flow Models (Flow Rates, Models, Preplanned Events)	76
9.3	Geographic Elements (Grid and Tile Points)	76
9.4	Data Element: Amp.....	77
9.5	Data Element: Attributes	78
9.6	Data Element: Bit_Count	78
9.7	Data Element: crc	78
9.8	Data Element: Day_of_Week	79
9.9	Data Element: Direction	79
9.10	Data Element: Envoke_Time	79
9.11	Data Element: Event_Type.....	79
9.12	Data Element: Fine_Time	80

SAE J2369 Issued MAR2000

9.13	Data Element: Flow	80
9.14	Data Element: Free_Flow_Rate	80
9.15	Data Element: Grid_Status	81
9.16	Data Element: Incident_Num	81
9.17	Data Element: Index	81
9.18	Data Element: Index_12	81
9.19	Data Element: Index_8	81
9.20	Data Element: InLine_Model	81
9.21	Data Element: Item_Byte_cnt	82
9.22	Data Element: Item_cnt	82
9.23	Data Element: Len_Time	82
9.24	Data Element: Issuing_Time.....	83
9.25	Data Element: Link_Speed	83
9.26	Data Element: Mapping_Seg	84
9.27	Data Element: Model_Shape	84
9.28	Data Element: Name_Field.....	84
9.29	Data Element: Occupancy	84
9.30	Data Element: Overlay	85
9.31	Data Element: Private	85
9.32	Data Element: Revision	85
9.33	Data Element: Scaling	86
9.34	Data Element: Start_Time	86
9.35	Data Element: T_F.....	86
9.36	Data Element: Technology	87
9.37	Data Element: The_Delta	87
9.38	Data Element: The_Grid	87
9.39	Data Element: The_LL	87
9.40	Data Element: The_Model	87
9.41	Data Element: The_Point.....	88
9.42	Data Element: The_String	88
9.43	Data Element: Vertical	88
9.44	Data Element: Way	88
9.45	Data Element: Zoom	89
9.46	Data Element: Byte[x]	89
10.	Aspects of Data Encoding (used in Transmission)	89
10.1	Practical Text Encoding Examples	90
10.2	Practical Text Letter Pair Encoding Examples.....	93
10.3	Local Table Encoding Considerations	94
10.4	Local Datum Encoding Considerations.....	96
11.	Aspects of Data Decoding (used in Receiving)	97
11.1	Grid Expansion Examples, Translating Out of the Grid Profile.....	97
11.2	Handling Incident Messages and Models	102
11.3	String Processing Examples.....	104
11.4	Letter Pair Expansion Examples.....	108
11.5	Examples of processing the CRC value for each packet.....	108
12.	Standardized National Tables Used in this Document.....	109
12.1	Modified ASCII Table	109
12.2	National Names Table	111
12.3	National Grid Table	123
12.4	Incident Modeling Shapes Definition Table	124
12.5	Incident-Event Table	126

13.	Guidelines for Standardized Testing of this Standard	126
Appendix A	List of National Table Entries and Encodings	127
Appendix B	Examples of String Handling in Messages	127

TABLE OF FIGURES

Figure 1	Major System Components	6
Figure 2	Typical Morning Commuter Link Impedance Model.....	13
Figure 3	Brute Force Multiple Messages versus a Predictive Model	14
Figure 4	Schema of Token Passing Process	16
Figure 5	Illustration of Various Data Packets From Each Sub-Stream	18
Figure 6	Use of the Four Schema Flows in a Typical Receiver	19
Figure 7	Conceptual Flow of Data Using the Grid Profile	27
Figure 8	Basic Model Shapes	31
Figure 9	Relationship of Incident Flow Model Message to Actual Model Entries.....	32
Figure 10	Model Use When Transmitted In-Line with Message	33
Figure 11	Models with an Added Element Due to an Incident	34
Figure 12	Comparison of a Models With a Negative Flow Rate	35
Figure 13	Simple Reuse Use of Models in a Traffic Flow Example	36
Figure 14	The Parts of a Message Packet.....	38
Figure 15	The Parts of a Data Frame	38
Figure 16	Detailed Context Information	39
Figure 17	Dynamic Range of Primitive Data Elements Compared	77
Figure 18	Road Topology of Segment used in the Example	97

TABLE OF TABLES

Table 1	Supported ATIS Message Types	12
Table 2	Planned Phase II ATIS Message Types	15
Table 3	Comparison of Protocols	15
Table 4	Background Concepts	17
Table 5	Schema Levels and Streams.....	18
Table 6	Primary and Auxiliary Tables of Names.....	21
Table 7	Types of Tables	23
Table 8	Example of Table Control Expansion Bits	23
Table 9	Example of Numeric Table Control Expansion Bits	23
Table 10	The Standard 32-Bit Lay/Long Values By Bit Value	29
Table 11	Scaled Values for LSB of the _Point with Zoom Settings	30
Table 12	Contents of a Model Entry	32
Table 13	Contents of a Model When Used In-Line	32
Table 14	List of Streams with Contexts	40
Table 15	Table Numbering Assignments.....	41
Table 16	Binary Data Context Flag Settings.....	42
Table 17	Binary Data Context Table Entries	42
Table 18	Expansion Tables Context Flag Settings.....	44
Table 19	Expansion Tables Context Table Entries.....	44
Table 20	Forward Referencing Context Flag Settings.....	45
Table 21	Forward Referencing Context Table Entries.....	45
Table 22	Grid Array Context Flag Settings	46
Table 23	Grid Array Context Table Entries	46
Table 24	Incident Flow Context Flag Settings	48
Table 25	Incident Flow Context Table Entries	50
Table 26	Efficient Incident Formatting	50

Table 27	Incident Modeling Context Flag Settings	51
Table 28	Incident Modeling Context Table Entries	52
Table 29	Incident Numbering Context Flag Settings	53
Table 30	Incident Numbering Context Table Entries	53
Table 31	Map Encoding Context Flag Settings	54
Table 32	Map Encoding Context Table Entries	55
Table 33	Private Data Context Flag Settings	55
Table 34	Private Data Context Table Entries	56
Table 35	RDS Incident Context Flag Settings	56
Table 36	RDS Incident Context Table Entries	56
Table 37	Resource Availability Context Flag Settings	56
Table 38	Resource Availability Context Table Entries	57
Table 39	Textual Incident Context Flag Settings	57
Table 40	Textual Incident Context Table Entries	57
Table 41	Transit Schedule Context Flag Settings	57
Table 42	Transit Schedule Context Table Entries	58
Table 43	Weather Context Flag Settings	58
Table 44	Weather Context Table Entries	58
Table 45	Vanity Map Flag Settings	58
Table 46	Vanity Map Context Table Entries	58
Table 47	Alternative Names Context Flag Settings	59
Table 48	Alternative Names Context Table Entries	59
Table 49	Tables of Tables Context Flag Settings	60
Table 50	Tables of Tables Context Table Entries	60
Table 51	Message Types	60
Table 52	Contents of a Model when used In-Line	68
Table 53	Contents of a Table Model Entry	75
Table 54	Contents of a Table Model Entry	76
Table 55	Contents of an In Line Model Entry	76
Table 58	Inline_Model Elements	81
Table 59	Occupancy Values	85
Table 60	Technology Types	87
Table 61	The_Model Elements	88
Table 62	Example of Encoding Non-Ideal Strings	96
Table 63	Example ATIS Message for Encoding	97
Table 64	Variations In Encoding a Segment Position in the Grid Profile	98
Table 65	Summary of Various Grid Exceeding Examples	102
Table 66	In-Line Data Elements	103
Table 67	Mapping the Message to the Data Elements	103
Table 68	Mapping the Message to a Model	103
Table 69	Modifying the Model Contents	104
Table 70	Using an Indexed Model for Complex Events	104
Table 71	Compound Contents of Tables	105
Table 72	Modified ASCII Entries for Letter Pair Selections	110
Table 73	Modified ASCII Entries for Table Selections	111
Table 74	First 171 Names In National Table	112
Table 75	Incident Term Names in National Table	112
Table 76	Proper Street Names in National Table	113
Table 77	National Grid Table Entries	124
Table 78	Incident Event Initial Table Entries	124
Table 79	Incident Event Initial Table Entries	126

1. **Scope**—This SAE Standard outlines the US standard for sending Advanced Traveler Information Services (ATIS) messages over various Bandwidth Reduced Media (BRM). It specifies the “over the air” format and method for such data messages at a bit level. The methodology allows a cooperative sharing of the media’s total bandwidth with other non-ATIS data services. It allows for further expansion to other ATIS messages such as transit schedules and weather use. It uses a common directory of messages to divide out those messages which are the subject of this specification from others.

It is built upon the concept of “waveform neutrality” in that it can be hosted by a wide variety of media often found in the delivery of wireless data services (although it is also useful for wire line applications). This message set is incorporated into each medium (“host medium”) by a formatting process specific to each media. A graphical overview of this process is shown in Figure 1. The capacity and services available over the host media can vary considerably in terms of capacity, other services present, one way or two way use, suitability for use, and a variety of commercial aspects not considered here.

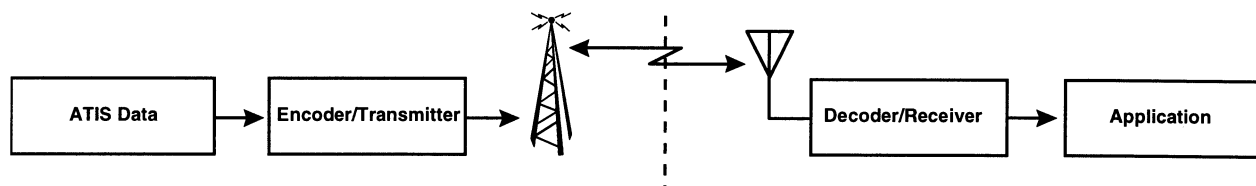


FIGURE 1—MAJOR SYSTEM COMPONENTS

This document is divided into five primary sections. The first provides a conceptual background of various elements required to understand the message encoding process itself (chapters 4, 5, 6). The second section contains precise definitions of the messages and data elements in the messages (chapters 7, 8, 9). The third section provides implementation advice and comments for both receiver and encoder manufacturers (chapters 10, 11). The fourth section contains tables of constants used in various message encodings of the standard (chapter 12). A 5th section, contained in the appendices, is an illustrative example of the message sets implemented with real data. Finally, a set of Appendices, SAE J2369-1 provides information regarding how to implement this document on a variety of host media.

2. References

- 2.1 **Applicable Publications**—The following publications form a part of the specification to the extent specified herein. Unless otherwise indicated, the latest revision of SAE publications shall apply.

SAE J2353—Data Dictionary for Advanced Traveler Information Systems (ATIS)
 SAE J2354—Message Sets for Advanced Traveler Information Systems (ATIS)
 SAE J2374—National Location Referencing Message Standard, Information Report

- 2.2 **Related Publications**—The following publications are provided for information purposes only and are not a required part of this document.

- 2.2.1 SAE PUBLICATIONS—Available from SAE, 400 Commonwealth Drive, Warrendale, PA 15096-0001.

SAE J1761—Information Report on ITS Terms and Definitions
 SAE J2355—ITS Data Bus Architecture Reference Model Inf. Report
 SAE J2366—ITS Data Bus Protocol Standard
 SAE J2367—ITS Data Bus Gateway Reference Design Recommended Practice

- 2.2.2 EIA - NAB PUBLICATIONS—Available from EIA, 2500 Wilson Blvd, Arlington, VA 22201-3834

United States RBDS Standard, Specification of the radio broadcast data systems (RBDS) August, 1997, DRAFT Version 2.

3. Definitions

- 3.1 Auxiliary Data**—The data found in auxiliary tables.
- 3.2 Auxiliary Table**—A type of table used to convey additional data not found in a primary table. Also called a secondary table. Typically, such tables are of the sparse type, but this is not a requirement.
- 3.3 Bandwidth**—Taken in the context of this document, the relative rate of information per unit time which can be sent under a given set of environmental conditions. Bandwidth is relatively limited in most mobile environments and hence effort must be made to conserve its use if the channel is to be used to maximum benefit to all parties. To the ATIS user, this means to be able to gain as much useful data as possible. To the operator/owner of the channel, it means to be able to serve more clients and hence increase profits.
- 3.4 Bandwidth-Reduced Media**—A term used in this document to refer to media in which the ability to send messages and information is limited due to a variety of possible reasons. This document was developed originally for use with FM SubCarriers, a bandwidth reduced environment where current state of the art limits successful delivery of data to rates under 10 KBPS. In context of this document, the term BRM is used in the general sense of all wireless applications (both one way and two way), as well as wire line applications such as low data rate modems.
- 3.5 BER**—Bit Error Rate
- 3.6 Context Token**—A token used in a stream to establish the presence or absence of a specific condition. A typical example is the use of the “maybe” condition whereby the context token state (one or zero in this case) determines whether the next sequence of bits represents whatever object the “maybe” flag refers to.
- 3.7 Context of a frame**—A section in the beginning of each message frame where a number of specific data element formats are established and various default data values are set. This is the descriptive part of the messages in the context of a “self describing” message schema. The context is also referred to as the header of the frame.
- 3.8 Context_Flags**—A section of the message frame which contains data spanning over the header which contains a specified number of flags each having meaning regarding the ways in which the remainder of the frame (the payload data) are to be interpreted. These flags establish the precise format of the messages in the payload area and refer to the context tables for specific default entries as needed.
- 3.9 Context_Tables**—A section of the frame header, which contains a sequence of values (entries), which are used to interrupt the data contained in the remainder of the frame (the payload area). These values may be default interpretation of the data, or values regarding how to correctly connect this payload of data with prior payload frames when the message (typically a table) extends over a single frame in length.
- 3.10 CRC**—Cyclic Redundancy Code. A code determined from a process derived over all the subject bits which yields a unique value for the pattern of bits processed. CRC codes are used for several reasons in this document, most often to characterize the values of the contents of a message with a unique code for ease in sorting or processing. Because FEC Code is handled in the lower layers of the protocol, the CRC is typically not used for error correction. A CRC16 is used which produces a unique value for any sequence of bits up to 2^{16} in length.
- 3.11 Dynamic Association**—The process of associating specific frames, modulations, and encoding methods with related sequences of data in the transmission coding. Dynamic association allows the encoder to dynamically change its allocation of bandwidth between various users and to relate the results of this process to receiving devices. Related to this process but beyond the scope of this document is the concept of accounting and billing of the final bandwidth allocation in the encoder.

- 3.12 Exception token**—A value used in decoding a stream of tokens which invokes a change in decoding behavior. A typical example would be a string of characters in which an exception token of the form “0x80” (which is not a valid string value) is found. This would indicate that the next two bytes should be interpreted as a table entry to be processed (and an exception to the norm of further characters).
- 3.13 FECCode**—Forward Error Correction Code, a general method of error correction used in the lower levels of the modulation of most transmission systems.
- 3.14 Flea Bits**—Because the message frame and the messages which fill it may not fit to exact capacity, there are typically a few bits left over in any message stream which are not part of the message itself but would otherwise be wasted. These bits are referred to as “flea bits” and are used by the encoder in any manner it wishes (and hence can be set to any value it desires). Because a knowledge of the message is required to determine the end of the message (and the start of the flea bits) this process can only be used for messages which are in the public domain (not privately encoded).
- 3.15 Frame**—See Message Frame.
- 3.16 Frame Count**—A value used to count which of a set of message frames is being sent (e.g., the nth frame) within the ATIS stream. This is NOT the same as a count of sub-frames, which is a term used in the lower levels some protocols and used in the directory stream assignments.
- 3.17 Grid Profile**—A term used to refer to the encoding of latitude and longitude in this specification and also defined in the LRMS SAE J2374 information report. The Grid Profile as defined here is extended beyond just the geometry portions of SAE J2374 to use sets of names and other values as needed to convey the location of segments without ambiguity. As a rule, the other profiles of the LRMS can be readily converted into and out of the Grid Profile if a set of basic datum quality rules have been established and followed.
- 3.18 Header, message frame**—The beginning of the message frame used to hold the context data.
- 3.19 In-Line Expansion**—The process, in the receiving device, of expanding a value from the transmitted token into an expanded (less compressed) value for further use. An example would be reconstructing full 32-bit values for latitude or longitude from the compressed Grid values sent. The term “in-line” refers to the methodology of expanding the compressed tokens as the message is passed on to the next stage of usage. Also see Token.
- 3.20 Incident Model Number**—The value of the model used by an incident event (an index number into the tables of models).
- 3.21 Latency**—The delay between the time an object is transmitted (begun to be transmitted) until it is received (fully received). Many modulations use the method of interleaving to spread bits of information over a period of time (referred to as the interleaver depth) as part of the error correction coding. The time required for all the information to be recovered adds a waveform specific delay (latency) of anywhere from 3 to >30 s to the recovery of such data.
- 3.22 Linear Predictive Coding (LPC)**—A methodology of information compression, which observes that many sequences of information vary over a smaller range than their complete value domain would allow. LPC uses this observation to compress information transmitted to obtain compression. This document makes liberal use of the general theory of LPC to affect the data compression methods.
- 3.23 Link Speed**—The (average) rate of speed over which an average traveler can transverse a given segment (link) of roadway.
- 3.24 Link Travel Time**—The (average) time interval over which an average traveler can transverse a given segment (link) of roadway.

- 3.25 Local Table**—The table of string entries (names and phrases) used locally to save on transmitted text. This table changes slowly as local needs require and is used (with the other tables) to expand passed strings (street names and free text) back into their original format.
- 3.26 Location Referencing** —The general area of specifying to multiple users with different applications, different underlying base maps, and various needs, an unambiguous geographic knowledge of location. In general, the LRMS SAE J2374 Information Report of the SAE forms the basis for a number of methods to achieve this within the ITS community. The so-called “Grid Profile,” which is developed in this document, is one of these.
- 3.27 Loop Sensor**—An inductive loop of wire placed in the roadway to detect the presence or absence of a vehicle. Loop sensors are one of a number of sensor technologies used by a traffic management center (TMC) to obtain information regarding the link travel times and speeds of various roadway segments. In the context of this document, loop sensors are translated into travel times and speed for use by a partnership of TMC/ISP, then combined into messages which are sent to the encoder to be translated into the over-the-air message specification of this document. Typically Loop Sensors data will be output by the TMC using the TMDDext message set standards, a related ITS standard.
- 3.28 LRMS**—Location Referencing Message Set. The formal name of SAE J2374 and a related body of research relating to methods to exchange locations between users in the ITS community. It consists of a number of “profiles” which outline message formats used to convey information. Among these is the “Grid Profile,” which is used and elaborated in this document and is intended for applications which need to combine the various other profiles into a bandwidth restricted application.
- 3.29 MER**—Message Error Rate. The rate of errors at the message level after correction by the FECC of the lower level channel errors. Every host media experiences the occurrence of a corrupted message at some time. Some media (typically two-way) have the ability to re-try or re-send such messages; other media do not. The overall success or failure of a given host can be compared to other alternatives by consideration of its MER.
- 3.30 Message Context**—That portion of the message frame (often also called the header) which establishes default values for expansion of the frame from which the subsequent data (the message payload) may be parsed.
- 3.31 Message Frame**—The envelope of the message, the frame is the outer layer of the message, typically a multi-byte long structure ATIS message. The actual frame length will vary depending on the host media from as few as 22 bytes to as much as 512 bytes in length.
- 3.32 Message**—In the most general definition, messages are the groups of coherent thoughts which are packaged together in a set of bytes called a frame. This term is applied in two different ways in this document. First, a message is taken as being the same as the message frame (containing a context and the payload area). Second, the payload of a frame contains one or more messages (each made up of tokens as specified in the context).
- 3.33 National Table**—The table of string entries (names) used by all receivers to save on transmitted text. This table rarely changes and is used (with the other tables, typically the local table) to expand passed strings (street names and free text) back into their original format. It is typically not transmitted over the media as it would not change as the user moved from one local operational area to another.
- 3.34 Off-hours**—Non-peak hours of operation for the host are typically late at night. These hours are often used to send support files and longer messages which conflict with bandwidth needs during daily operations. This may prove a way to distribute small maps for self contained users. On some hosts, this may also be a time slot when higher speed modulation can be employed based on the pre-established stationary nature of the receiver during this time period.

- 3.35 Ordinal Post Name**—That part of the name appended to the “number” field of a numerical street designation, the “rd” in 3rd street or the “th” in 123th street.
- 3.36 Payload, of a frame**—The portion of a frame which contains the actual data and is used in conjunction with the context and tables to expand into a more complex message for use by the receiving device.
- 3.37 Post-Name**—The portion of the name field at the end of the primary name, such as *Street* in “North Main Street.”
- 3.38 Pre-Name**—The portion of the name field at the beginning before the primary name, such as *North* in “North Main Street.”
- 3.39 Primary Name**—The portion of the name field containing the primary name, such as *Main* in “North Main Street.”
- 3.40 Primary Table**—A table which is used as the “first entry” for table look-ups. Such tables are typically national or local in scope and do not change very often. They can be used in conjunction with secondary tables to provide a “one of many” look up function, however, the primary table entry is taken as the preferred value. For example, the primary tables might contain “Main Street” as an entry while the secondary tables provide a value such as “Hwy 39” and thus provide a local alias.
- 3.41 Raw Symbols**—The smallest data element found in the payload of the message frame. They can vary in length from one to many bits and have no octet alignment properties. Symbols are parsed (using the context and tables) to create values which fit into byte aligned memory attributes in receiving devices.
- 3.42 RDS-TMC**—Radio Data System—Traffic Message Channel, the proper term for the delivery of incident text messages over the RDS-RBDS modulation format. This format has been deployed in various ways over the low speed subcarrier in the EC and to a small degree in the US (RDS an 1187-bit/s raw data rate stream with ~200 useable bits per second). The message format used consists of hard coded locations (by index value and therefore requiring a common agreed upon map) combined with textual messages (which can be expanded into the local language of choice). There are no link impedance concepts in the RDS messages, only short textual descriptions. The text look up tables are not that different from the method used here, however, they are never transmitted over the media and do not change or evolve once issued.
- 3.43 Retransmission Time**—The stream of data found in some broadcast (one-way) hosts containing messages which are repeated frequently. This is referred to as the retransmission time and can vary due to several considerations including policy decisions, other needs for bandwidth and how often the data itself changes. Messages are also retransmitted to ensure that a user who may have missed a prior message can successfully decode it when repeated this type of time diversity improves the system robustness. Users are also entering and leaving the systems (turning their receivers on and off) at all times so considerations must be given to the population of “new” users who may not yet be tracking current data transmitted in prior messages. Two-way media will typically transmit on demand in a point to point fashion.
- 3.44 Route Impedance**—An electrical engineering construct commonly used to model the traffic flow rates. A high impedance implies a “slow” link. This term and *route travel time* are often used interchangeably. At a gross level, *rate of travel* multiplied by *length of segment* can be considered the same as the segment travel time.
- 3.45 Segment**—A portion of roadway network between two end nodes. Typically, one cannot “leave” the segment and enter another segment except at a joining node. See SAE J2374 for a more complete taxonomy of network description terms.
- 3.46 Services Directory**—A stream of information informing the user as to what information is available in the host data stream.

- 3.47 Sizing Data**—Information about a stream of data, typically a table being sent, regarding its ultimate size in transmitted bits and (in some message) in expanded bytes. This information is provided so that the receiver can determine if it has the capacity to recover the entire message. This may require the receiver to purge other information in its memory, the process of which is beyond the scope of this document.
- 3.48 Stream**—The concept of an independent flow of information multiplexed (in time) with other streams in the host data flow to effectively divide the bandwidth of the system into useful subset parts. Each stream tends to be a self-contained set of information. Some streams, which are supportive in nature, need only be downloaded once and then this information is used with other streams to decode the information. The transmission provider can decide what mix of streams best meet his customer needs in a manner similar to the way a broadcaster determines how often to broadcast the updating of voice traffic news.
- 3.49 S-T-T-S**—The four level schema used in the division of data into related streams. STTS refers to Sizing, Tokens, Tables, and Static stream information and it often denotes the process steps followed to receive a stream of data (of which ATIS is but one possible collection of data present on any given host). The process grossly is as follows. The sizing data in the directory must be examined to confirm that the device can handle the tables it will need. The tokens can then be extracted from messages to monitor the stream and get the subject data - this step is repeated as long as the data is desired. In use, the receiving device is must have (or get if it does not have it) the table data which is also periodically transmitted. And finally, if there are static tables which may not be transmitted, such as a National Table, it must have those as well - typically these are obtained from whatever source provided the decoding software in the first place.
- 3.50 Sub-String**—A term for the occurrence of one string inside another, and specifically the optimal matching of table entries to utilize this effect. For example "Road Construction Work" might be a locally used string. In this string are other National Table strings such as "road" which should be used to represent it. This is encouraged to save space. The governing rule is that a table can not 'expand' within itself. A consideration for encoder makers is to determine algorithms, which will pick the most efficient (typically longest) table entries rather than several shorter ones.
- 3.51 Table Extension**—The process of adding additional entries onto a previously issued table. This allows users with older tables (as determined by the revision number) to continue to use their current tables while obtaining a new one (with the new entries on the end). Encoders in general extend tables rather than renumber them (which must still be periodically done). The process of renumbering implies that the table has been resorted.
- 3.52 "Throw an exception"**—Programming jargon for placing into the stream of parsed token objects, a symbol which indicates some exception process must be made in the line of code to handle it. A typical example would be a stream of ASCII characters representing some text when the next character in order was a non-valid printing character such as 0x80. Detecting this character "throws the exception" and indicates that the next two characters will be taken together to form an index value into a table which will provide a text string to be expanded in line and replace the exception character.
- 3.53 Token**—The next object to be processed in a serial stream of data being parsed for expansion. It may be from one to N bits in length. The specific size can be determined by the message specification and the context flags set.
- 3.54 Topographical**—referring to the *height* of the subject geographic object.
- 3.55 Topological**—Referring to the *connectivity* of the subject geographic object.
- 3.56 "Toss Back Token"**—Programming jargon for an exception token appearing in an unusual place where it is typically (from statistical point of view) not found. Its presence is used to indicate that the expected exception WILL NOT OCCUR and that the 'normal' flow of symbols will follow. In many instances, it is not required to first place an exception token in the stream if the message format can be used to establish that a specific token will be the index value. For example, one of the ways to set the context flags in some messages uses the name

field of the segment as an index value (rather than a stream of ASCII which could contain such a value preceded by an exception). If the subsequent name entry was NOT an index value, the presence of the toss back token indicates that a stream of ASCII will follow instead. The use of this construct is almost always the result of a compression selection which failed in the specific case and resulted in more tokens being sent - this is acceptable as long as in the sum of all messages the compression choice was the correct one.

- 3.57 Transmission Epoch**—A period of time (TBD) where the information of the various streams is repeated. Also see epoch. This occurs typically in one-way broadcast transmission of data.
- 3.58 Variant Flag**—The lower two bits of a table index used to control how strings from that table are expanded.
- 3.59 Yellow Pages** —This term is no longer used in this document. Originally, it was defined as a portion of the Services Directory.
- 3.60 Zoom Scale**—The scaling factor used in the description of the Grid tile entries. A zoom scale can be used to focus the tile on a much larger or much smaller area of interest. The intended use of the zoom scale is threefold. To reflect weather and other events on a national scale where final precision needs not be very great, and to establish smaller regions of interest as might be found in an amusement park, or a hotel kiosk where the host media would be used for transmission of local data regarding these environments. A final use of the zoom scale is to create “vanity maps” over portions of the metro area which cover information in a non-geographically correct way to the benefit of the end user. An example of such a map is the local subway or transit system where a long established local graphic might be used in place of a geographically correct base map for reasons of clarity and tradition (the Washington, D.C. Metro system is an example).
- 4. Overview**—This document defines the message set for the transmission of ATIS messages over bandwidth reduced media (BRM). This chapter of the document provides an overview of the type of messages supported. This is followed by a review of the general conceptual protocol stack involved and some relevant aspects of the modulation, which are not part of this document but which are of interest in typical end user receiving devices.

This chapter provides overview material of a general nature concerning ATIS implementations on any media and a summary of the expectations for various network layers and other messages. It is written to provide a suitable background for both technical and non-technical readers with only a few references to bit level details. In general it presumes little knowledge about the chosen host media; however the supporting appendices for each media expect specific expertise in the media in question. Non-implementers wishing to gain a technical overview of the document should read Sections 4 and 5. Implementers will be most interested in Sections 6, 7, and 8 and the implementation guidelines chapters.

- 4.1 Message Types Supported**—The initial release of the document has an emphasis on supporting incident events as well as establishing the decoding process flow of data in this media. The following ATIS messages are supported using the basic incident model structure (see Table 1):

TABLE 1—SUPPORTED ATIS MESSAGE TYPES

Value	Message Type or Category
1.	Current (Instant) Link Travel Times
2.	Modeled (Historical) Link Travel Times
3.	Predicted (Future) Link Travel Times
4.	Un-Planned Incidents & Events
5.	Pre-Planned Incidents & Events
6.	Full Text Fields of Events
7.	Message Phrases for Events (a la RDS-TMC)

The earliest deployed traffic data channels emphasized presenting the user with a textual description of each incident event while providing little or no insight into the effect on the transportation network or the individual traveler. A typical example would be an RDS-TMC message such as "Overturned Lorry, expect 20 minute slowdown" which would be periodically repeated.¹ There were no predictive models of flow rates and few provisions for advance warning of pre-planned incidents. As such, these systems reduced the usefulness of the information to telling the traveler *why* they were already stuck in traffic rather than how to avoid it to begin with, or allowing the traveler to determine the best course of individual action.

In the current document, the emphasis is to provide the traveler with knowledge of the transportation network flow rates at all times - both actual and those which can be anticipated in the future based on prior knowledge and experience. The traveler, aware of the desired route/destination, can use this information to best traverse the network. The primitive information of incident texts (both those in progress and those which are known ahead of time such a construction or a major traffic event) are available as well. In addition, the traveler can use the data to ask *what if* questions such as what time he should leave to arrive at the airport across town based on anticipated travel conditions at that time. The models employed are shared and possess the property of super-position to allow reuse and further conservation of bandwidth. See Figure 2.

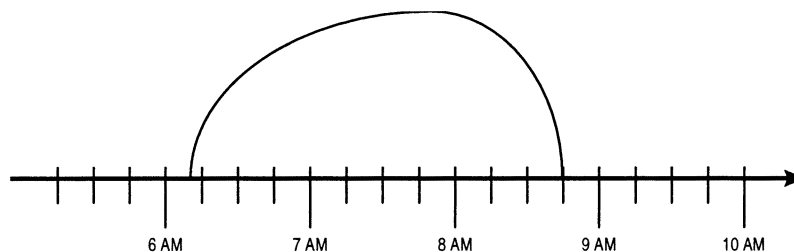


FIGURE 2—TYPICAL MORNING COMMUTER LINK IMPEDANCE MODEL

The messages in the prior list all use a common set of incident models to portray link impedance in terms of start time, duration, and magnitude. The use of these models will vary in practice based on the skill of the implementers, the availability of data, and the effort put forth to develop and maintain valid historical models. Coefficients for various model shapes and sizes exist - including one which simply sets the travel time at a specific rate until further notice.

At its most simplistic level of implementation, the models can be used to repeatedly update the current segment flow rates ("brute force") as is typically done in early deployment cities today.² In a more elegant implementation, expected daily flow rate models are augmented in real time by models reflecting the unplanned incidents of each day. In Figure 3, a single model message is compared with 16 messages³ used to model the same event. Because the models allow grouping and super position, the changes from an "average" day's delay can be readily added and updated as incidents and predictions change over time.⁴

1. A further problem of the RDS-TMC based implementation was that road segments were numbered as hard coded values. This produced a system wherein all parties had to use a similar map based on the same location reference numbering. This is a major problem with multiple map vendors who will not agree on a common ideal map. As a result of this issue and others, the American LRMS effort has devised a method, which uses the actual lat/long values of key segments to exchange a single datum across multiple maps. This work has been adopted as the US standard for location referencing and is SAE J2374.
2. The development of modeled flows is produced by processing incident free "average" days over a sliding window period of time to produce a low pass filtered model which varies with the time of day. Such models are self maintaining in that they evolve and change as traffic flow patterns pass through the encoder stages. It is anticipated that most cities will develop such models over a period of time when adopting the standard, starting initially with only real time flow rates. The document promotes this type of evolutionary growth without requiring changing in the receiving device software (and therefore promoting earlier national commercial deployment). In smaller metropolitan areas where the number of monitored street segments will remain small, sending only the current data is possible from a bandwidth perspective. However, this should be avoided as it places a higher processing burden on the receiving device and does not allow the device to make predictive estimates of traffic flow.
3. Over the 2.5 h event shown in the figure, there would be 30 messages if a message was sent every 5 min. By contrast, if this flow was a normal one the end user would probably have received the model the pervious day and need not be in contact with the host media to have an average predictive insight into this segment. In order to support the needs of new users continually entering the network, this model would be periodically transmitted as well. So the message savings, while significant, is not the full 30:1.

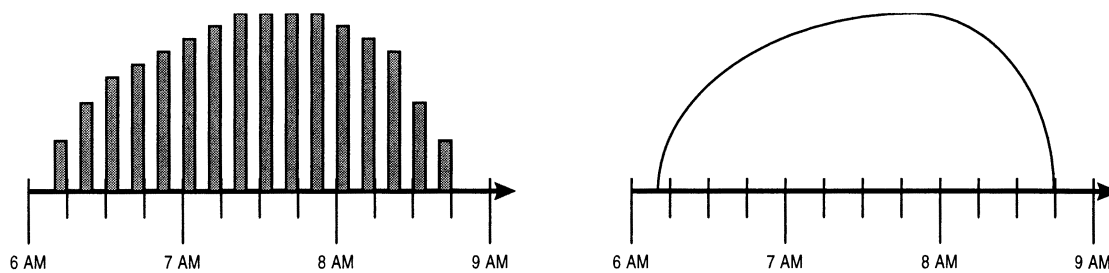


FIGURE 3—BRUTE FORCE MULTIPLE MESSAGES VERSUS A PREDICTIVE MODEL

A major goal of the document is that it must support the previous messages across different traffic management centers sources, different map databases, coverage areas, and a range of end user devices as well as supporting the efficient automated encoding goals. To achieve this, a very basic map is also transmitted which contains the *from-to* endpoints and names of the street segments of the local metropolitan area.⁵ This map database can be readily matched to much higher quality databases which end user devices are expected to contain, or it can be used by itself in very low end applications.⁶ The messages used to transmit the map can also be utilized to transmit map fragments when instructions for re-routing of traffic need to be provided.

In addition to the previous ATIS messages, there are a number of supporting messages which are a part of the general message structure for the media (that is, they are used in other data stream besides ATIS). These are basically messages for sending compact tables for expansion use contained in the "streams" of message frames described as follows. The term stream refers to a related sequence of messages, which make up a group of related messages, such as ATIS.

Other streams of data useful to ITS applications may also exist on the specific host media. The format of these messages is beyond the scope of the present document. In addition, there may be one or more private *for pay* data streams which may also be co-resident on any particular host media. The receiving device would examine whatever format of a service directory existed in the host media to determine what other services and messages can be found on it.

In the next stage of the ATIS standards development for bandwidth reduced media messages, the emphasis will be on multi-modal transportation messages. This will allow a common message structure for transit routing and schedule adherence messages as well as occupancy and capacity data. The messages which will be supported include (see Table 2):

4. Readers who are practitioners of traffic management will please forgive the use of the terms "average" and "typical" when referring to traffic flow patterns in this document.
5. Calling this a map is perhaps gratuitous. Its use is limited to matching to higher quality maps or as a rudimentary display of traffic conditions. It does not contain segment shape points, address ranges, or any routing data.
6. Methodologies and recommended practices for doing so are being developed and tested by the Oak Ridge National Labs (ORNL) project on location referencing.

TABLE 2—PLANNED PHASE II ATIS MESSAGE TYPES

Value	Message Type or Category
1.	Parking Lot Availability/Status
2.	Ride Share Availability/Status
3.	Transit Schedules (for Bus, Rail, Airline and others)
4.	Current Schedule Adherence (for Bus, Rail, Airline and others)
5.	Weather Events (current and predicted)
6.	Other "non roadway" local events

Most of these messages have to do with schedules of pre-planned routing combined with real time adherence to the plan. In addition rudimentary maps augmenting the routes covered by these services will be provided using the methodologies described in this document.

4.2 The Protocol Stack—This document describes a set of messages, which reside as the "application level" on a protocol stack. The underlying levels are provided by the host media. This specification provides the bit level constructs for a set of messages which are in turn framed, encoded, and recovered by the methods of the host media. This process is typically a part of a standard for that media. The specific number of layers and processes varies with the host, however the figure below is illustrative of the process. Those host media which are two-way in nature, tend to support a retry mechanism in the delivery of data, while those that are one-way, tend to support a datagram like environment. See Table 3.

TABLE 3—COMPARISON OF PROTOCOLS

OSI Label	Technical Overview	Equivalent
Application	Distributed information Services	J2369
Network	Stream Directory Frames	Specific to each Host
Data Link	Block and Frame FECC	Host Waveform
Physical	A Modulation Format	Host Waveform

In this analogy, SAE J2369 provides the application layer that ATIS information upon which end vendors build useful applications.⁷

4.3 Physical Modulation—The physical modulation of the waveform shall be as defined by the specific host media. Other source documents which will in turn provide information on the physical modulation, data link layer, network link layers, and any other required information needed to use this media as a host.

Implementers of this document can in theory ignore these issues. Conceptually, users of this document need not be aware of the modulation used, but can rely on the host to deliver messages in the SAE J2369 format. In actual practice some understanding of the framing and delivery structures is expected to be required in order to re-assemble or request messages from the lower layers of the host media.

4.4 The Network Layer—In order to select which data frames (packets of data) the user desires, and to logically connect these frames (both in terms of the encoding used and in functional relationships of the various streams) information about the formats provided by the host of interest is required. We refer to this mapping of information as *the network layer* in this document, but the reader is warned that a variety of conflicting terminology exists in the documentation found for various media. The purpose of the network layer information is to be able to determine what information within the host media is that representing the messages of this document.

7. Multiple streams of other application layers can typically be processed by the same receiver device at the same time. This allows multiple applications to co-exist over the same media and in the same receiving device, limited only by media bandwidth.

Note that the length of such packets is fixed in some media while it varies in others. This document allows for lengths from about ~20 bytes up to 512 bytes. Effective use of bandwidth is achieved by packaging the maximum number of payload items into each data frame offered by the host media. At other times, the data issuer may desire to organize messages with "one ATIS message per packet" for commercial reasons. In cases of very restricted bandwidth, the host media may break down the data frame, extract the context portion and then send the payload in very small clumps to be reassembled as needed. All of these variations are legal and are handled with generally similar and reusable software algorithms at the receiver end. Consult the Appendices for specific examples.

The transmission epoch (for one-way broadcasts such as commercial FM systems, some pagers, and some beacon systems) also varies with the host chosen and what other use of the bandwidth these messages must co-exist with. It is expected that most hosts will employ time interleaving techniques to support multiple users. In the case of two-way use of this document such a repeating epoch may or may not exist depending on both the media and the operators needs (e.g., CDPD, GSM phones, and AMP phones each allow the concurrent delivery of broadcast messages along with point-to-point messages). Most two-way media enjoy the added advantage of being able to request specific information (in this case specific messages) from the information service provider, a bandwidth management ability which one-way transmissions do not typically have.

4.5 Data Tokens and Stream—Already used a number of times in the text are the terms *message*, *token*, *table*, and *streams*, this section provides an overview of these terms before a complete definition is provided in the next section. Basically the ATIS type *messages* in this document come packed into a message frame of a variable length number of bytes which is made up of one or more (typically related) messages. The data elements of these messages are referred to as *tokens*. Tokens are the compact building blocks sent over the media itself, rather than sending a more complete verbose message. Tokens are generally used to index into *tables* (which are themselves sent in other message frames, but with less frequency)⁸ to expand the message back into a more useful and verbose format. The essence of the data compression used in this document is achieved by sending these compact tokens over the media. See Figure 4.

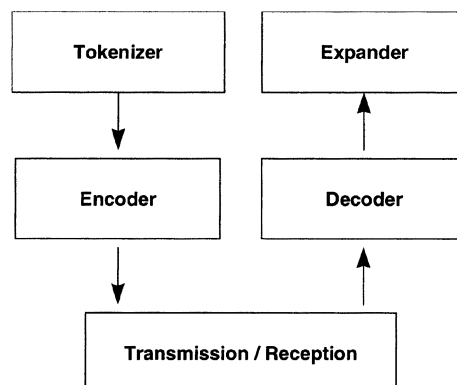


FIGURE 4—SCHEMA OF TOKEN PASSING PROCESS

The host media can be viewed as a river of message frames. Each of these frames may relate to different data services and needs supported by the host. In this document, we are concerned with those dealing with ATIS. Within each frame are one or more messages, which in turn contain the tokens and tables.

8. In cases of host media with extreme bandwidth limitation, it shall be allowable to send the supporting tables by other media than that of the host. By this method very limited devices can be supported, but the ability to dynamically upgrade tables is all but lost (i.e., table becomes priori agreed upon and static in nature much like an RDS system). As a result, this technical approach is not recommended if the media could support a more dynamic delivery of the tables.

A simple view of the transmission process is as follows: A two-step procedure at the transmission end starts with the tokenizer which converts the data into tokens⁹, followed by the encoder which combines the data with other material and adds the complex error correction codes. This is then modulated along with other signals out over the air by the host media transmitter. The signal is recovered by the end user receiver and converted back to raw symbols. These symbols are then decoded (error correction is applied to ensure correct data being received) to recover the message frames (and the tokens within). Finally, in an expansion section, the tokenization process is reversed to produce a complete message. Note that we do not say “the original message” as that form is probably not of use to the end receiver device. Typically, the receiver device will use other message protocols such as SAE J2353 and J2354 or merge the messages directly into whatever proprietary data structures are being used.

From the previous, it is also evident that a SAE J2369 receiver is more than simply a “bent pipe” device in that it must maintain sufficient state memory to perform both the decoding of the raw symbols and the table lookups needed by the token process.

4.6 The Examples Used in this Document—This document makes use of non-normative examples, which are taken from real work datum gathered from various ITS programs and other publicly available data. The appendix sections contain a small example set of geographic data that has been encoded in the Grid and names table stream defined in the document. The data used for this example was US Census Department TIGER data. This is presented as a non-normative example as part of the implementation guidelines of the document.

5. Background Concepts—As stated in the previous section, the need to conserve bandwidth has led to a methodology wherein streams of “tokenized” data are expanded in the user receiver to be used by the end application. There are several key processes at work in the document to insure the efficient use of media bandwidth, a theory of operation for each is provided in this section. The basic concepts include (see Table 4):

TABLE 4—BACKGROUND CONCEPTS

	Method	Subject
1.	Stream Schemas	How the information streams work
2.	Tokens and Tables	The theory of data compression used
3.	Tiles & Grids	Efficient representations of latitude & longitude values
4.	Traffic Models	Shared models of link impedance modeling
5.	Forward Pointers	Redirection to other data sources
6.	Frame Bit Stuffing	Unused “Flea Bits” in the message frame

These methods are uniformly applied throughout the process of SAE J2369 messages and also used with other streams of data not described here (for example, a stock quote stream uses similar table expansion). The theory of each is explained in turn in this section. The specific bit level encodings are described in detail in the data element section.

5.1 Stream Schemas—How the Information Streams Work—The single logical ATIS information stream is broken up into four of sub-streams, each contained in their own logical data stream which is in turn multiplexed into the final physical data stream of packets along with other information. For purposes of explanation, we will adopt a broadcasters mindset of the host media. Bear in mind that a two-way media would function similarly; however, the reception of specific messages would be in response to requests rather than in a simple time rotation. Using a graphic of the river of data packets, this can be shown as Figure 5:

9. Note that this document specifies the over-the-air message format, it does not specify a standard interface between the source of the ATIS messages and the encoder itself although significant illustrative portions of the document deal with conversion issues and implementation comments from various other common formats.

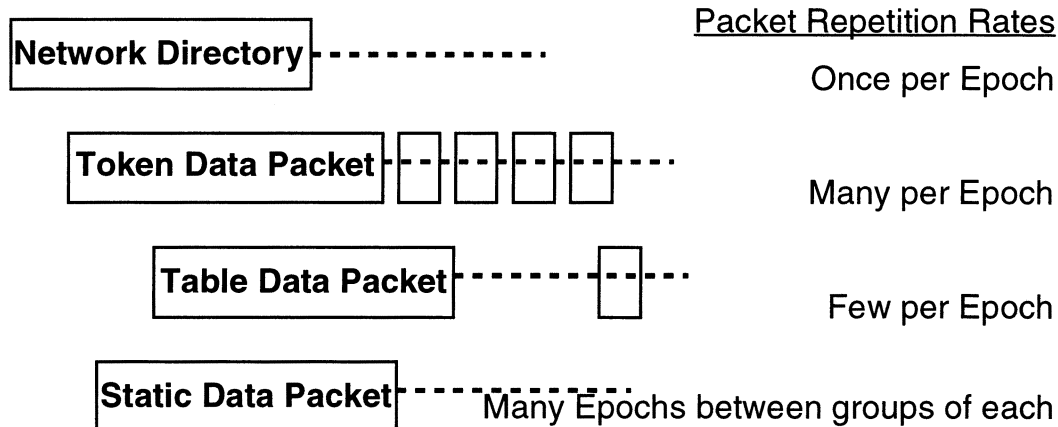


FIGURE 5—ILLUSTRATION OF VARIOUS DATA PACKETS FROM EACH SUB-STREAM

A four level schema model is used in developing these information (data) streams for transmission. The four levels are listed Table 5. Taken as a whole, these four streams provide all the information required to fully expand the data to the maximum point of usefulness. This four level schema is referred to as “S-T-T-S” in the text.

TABLE 5—SCHEMA LEVELS AND STREAMS

Schema Level:	Found In:	Containing
Sizing Data	Directory Stream	Tables Sizes, Global Data
Token Data	Main Stream	Sets of Contexts & Tokens
Table Data	Main Stream	Sets of Dynamic and Local Tables
Static Data	Off-Stream	Sets of 'Static' Data and Tables

5.1.1 **SIZING DATA**—Sizing data is contained in the directory stream transmitted periodically in the network directory packets and in the service directory. This level of the schema provides the following information to the receiving device:

- The current revision of each required table and their interconnections
- The size of each table itself in octets
- The max size of any object expanded from the table in octets
- The expected lifetime of the table

In addition, when streams are about to be transmitted within the current directory epoch, information regarding when the tables and token streams will be transmitted is also sent (this data is dynamically allocated on a local basis and can change from network epoch to epoch). This data includes the precise frame counts when each of the other levels of the schema will be transmitted. This data is then used to select which frames should be captured and processed further. When a table is not to be transmitted during the current epoch (as is typically the case with the static tables) then the services directory will contain data regarding what future epoch in which the stream will begin transmission (if this is known). In this way, the receiver can know in advance when periodic tables will be sent. See Figure 6.

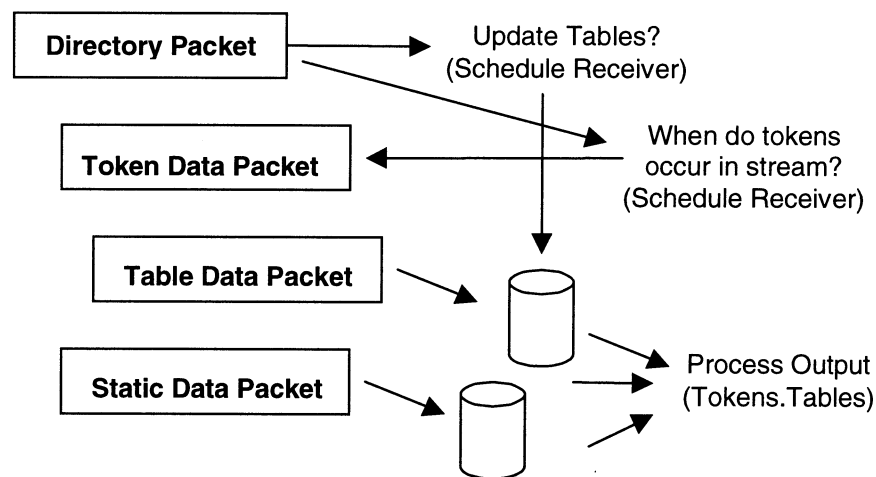


FIGURE 6—USE OF THE FOUR SCHEMA FLOWS IN A TYPICAL RECEIVER

In the receiving device, the sizing data is extracted from the packet of the services directory and the version of the needed tables is compared with that currently loaded into the device. By this means, the device becomes aware of the need to periodically update from the two streams of tables. The receiving device can also determine how much memory is required to process the subject information stream after expansion occurs and therefore better manage its memory space¹⁰. When the sizing data indicates that new table data is needed, the directory stream is compared to locate which stream contains the data required. These tables are expected to be broadcast often (every several minutes when changing with dynamic data). When static table changes are required (much less often, perhaps monthly¹¹), they will typically be scheduled for daily off-hour transmission. The service directory will have an entry, which can be used by the receiving device to know when to tune in. See the static data table discussion as follows.

In summary, sizing data which is part of the services directory, is used to detect the need to load other levels of the schema and is periodically checked as a function of recovering the directory data used to track the token stream.

- 5.1.2 **TOKEN DATA**—Token data is the stream of data which is processed by each end user device (typically in real time). Token data is highly encoded and compressed and can only be utilized after expansion (the section on tokens and tables as follows, discusses compression methods). In the “normal” operational mode of use the end user is concerned with only the token and table logical data streams which appear in short cyclic rotations of the transmitted data. Before this can occur, the device will have looked at the sizing data and loaded the proper tables for decoding. After this point, the receiver need only handle the tokens as they occur, using them to generate whatever final messages or state data the end application implements. Receivers may process other data, sleep cycle, or perform other functions between periods of token packets¹². An example of token data would be a stream of tokens which updated the current impedance of several street segments. Tokens can be expanded based on either, both, or on one of the two table types; this is dependent on the token implementation.

10. A memory limited device may not be able to successfully download and expand all data types which are available to it over all the different host streams which it can receive, hence the end user may have to make conscious choices regarding which data tables to delete to make room for new data. On a more subtle note, a user device may wish to begin the process of downloading tables in anticipation of a need for decoding with them, consider the case of a user traveling across a very large metro area.

11. An example would be a basic map update of the local area.

12. In some hosts, the precise period of packet arrival is known from the frame counts in the directory stream data; in other hosts, this is periodically transmitted.

- 5.1.3 **TABLE DATA**—Table data is the stream of table look up data used in conjunction with the token data to expand the tokens into a more useable format. While table data changes on a regular basis, it is constant for a notable period of time¹³, and therefore the data stream need not be monitored by the end receiver device at all times. An example of table data would be a set of entries comprising a shared model for street impedance versus time of day. Table data can change without warning or consideration for tables made outdated by the issue of a new table, whereas provisions are made for the controlled updating of static data tables.
- 5.1.4 **STATIC DATA**—Static data tables are so named because they are expected to change only occasionally and for major reasons. This would involve such things as the national naming table or the name table for a local city. Another example would be an underlying common base map used for token expansion. While the static data tables change infrequently, they will still tend to be transmitted daily during off-hours periods. Smaller data tables, for example a very high level base map of the area, may be transmitted during peak-hours as required¹⁴. Hence, one would expect to see a transmission time posted in the yellow pages directory data stream. Like all tables, changes are posted before they occur in the sizing data so that graceful changeover can occur. When possible, an interim table which expands prior entries should be used (and the proper bits in the sizing table set) rather than issuing a new table which makes the now obsolete table non-functional and might therefore strand a portion of the user base. The sizing data contains information on when a table will expire, if the new table will allow overlap operations and when it or the new table will be next slated for transmission (in the service directory).

In addition, these tables may be so static and universal in nature that the receiving device may be able to obtain them from other sources¹⁵ upon demand rather than have to be operational during off-hours periods. In addition, customized tables developed by specific vendors with various proprietary improvements may be available only from such other sources. An example of such a table might be the national naming table augmented with information for voice synthesis or a map database suitable for routing, etc.

A typical table structure is the *Primary Table* (Pri_Tbl) made up of a sequence of table entries (their index position imputed by the order) where the table entries are in the localized version of a character set.¹⁶ Internationally this would be modified Unicode values. Here in the US where ASCII is the primary Latin character set, we have developed two modified variants of ASCII where the “non-printing” upper 127 characters are used in a Huffman-like compression of character set values. The first encodes the most common dual-characters values (ea, ei, mm, etc.) the second encodes on the most common street name tokens (Street, Drive, etc.). In use an exception character (0x80) is first thrown to access the tables. Strings expand in a two-step process, first replacing the index value with the table entry contents which is made up of modified ASCII, and then replacing modified ASCII with a full Latin-1 character set.

-
13. A reasonable question would be how long is the data good for, however this will vary with the host media, local conditions, and encoder implementation. The structure of the encoder allows for a dynamic updating of the tables to add new entries as they become useful and remove old ones. Under such conditions, the table could change hourly. A less fully featured encoder's table might only change every few months. Transmission rotation rates of the tables can also be selected locally and dynamically so as to fit the needs of local users.
14. Such a map might serve a traveler arriving by plane with a personal device until such time as a more complete map could be downloaded. The selection of such rotation rates is seen as a local choice and can be used by the information provider to differentiate his offerings, hence this document does not mandate such a choice, rather it gives recommended settings in the implementation sections.
15. The Internet, local Kiosks, other communications media etc. One of the requirements placed on the encoder is that it support the ability to publish to others any static tables which it creates. Some static tables, which are national in scope and usage, are provided to the encoder developer and to end equipment developers by the SAE.
16. Table entries can point to combination of basic symbols and to subsequent tables in a nested fashion. The most basic symbols are not transmitted as “ASCII” but rather as a compressed modified ASCII format for further savings. Implementers can store messages in this format if they are memory limited or may chose to expand tables entries when stored, therefore eliminating the need for expansion routines when called. In any event, the basic symbol set is a national level table (a static table in the STTS schema) and is not subject to change on a local level.

Tables also allow for the concept of super-position by the use of an overlay table structure referred to as an *Auxiliary Table* (Aux_Tbl). Consider the naming needs of a segment with one or more aliases. In the example map, there is a freeway segment called “I-10 Freeway” which is also known by the alias “San Bernardino Freeway” and by the alias “Santa Monica Freeway” depending on the segment selected. Thus each of the three possible name sets for segments which are a part of this freeway are supported (index values 4,5,6). The auxiliary table allows these secondary names to be established in the data flow and then used or not at the receiving end. Note that by this method, different primary names can be selected for different segments as well. See Table 6.

**TABLE 6—PRIMARY AND AUXILIARY TABLES OF NAMES
(SHOWN WITH FULL ASCII TEXT FOR CLARITY,
IN PRACTICE ENCODED SYMBOLS WOULD BE USED)**

Index	Primary Table	Auxiliary Table
0	0x0000 (Toss back Token)	
1	Dog Street	
2	Cat Drive	
3	Frog Avenue	
4	I-10 Freeway	
5	I-10 Freeway	(5) San Bernardino Freeway
6	I-10 Freeway	(6) Santa Monica Freeway
7	Main Street	

The structure of the Auxiliary Table is a sequence of sets of: [an index value followed by the entry]. Whereas primary tables have an imputed index value which advances by one for each entry, the sparse nature of the auxiliary tables require an explicit index value. Any number of auxiliary tables can be associated with another table. A related basic construct is the Auxiliary Data Table (Aux_Dat) which allows an arbitrary sequence of data elements to be associated with a single table index entry.

In summary, the primary real-time data is the stream of tokens and tables, which can be located based on entries in the directory stream. Embedded in these entries is the sizing data, which can be monitored to detect changes in the revisions used in the tables themselves. Tables that are static for long periods of time are transmitted off-hours or may be obtained from other sources and are in turn used to expand the tokens into more useful formats. Strings of text are restored to an uncompressed format by a two step expansion process.

5.2 Tokens and Tables—The Theory of Data Compression Used—The message set uses a few data compression methods uniformly on almost all data, which is transmitted to achieve the highest possible effective bandwidth. These methods result in an on average saving of over 13x as compared with uncompressed transmissions (or an equivalent data rate of over 100 Kbps). The process chosen is lossless¹⁷ and easily reversible with the least effort to reconstitute the data at the received end. Basically we exploit our *a priori* knowledge of the data stream and the concepts of entropy to reduce the symbols transmitted to be the smallest tokens possible, then expand these tokens at the receiver.

The class of data compression used is generically referred to as Linear Predictive Coding (LPC). More specifically, a combination of breaking up the input stream into fixed length tokens (as in Huffman coding) and into variable length streams (as in Ziv-Lempel)¹⁸ is used on the geographical point and the names of streets and phrases. Peter Gutmann, in a paper titled *Huffman and Related Compression Techniques* coins the excellent phrase “substitutional compressors” for this type of algorithm. Added to the compression methods is the general concept of deviations from a modeled behavior for segments when such data is available. Finally,

17. Apart from the initial truncation of any bits deemed to be of excessive precision.

18. However, unlike selected forms of LZW-LZC, we do not effect transmitting the data dictionary (tables in this document) to the decoder as is done in common PC and UNIX implementations such as PKzip. We do make provisions in the encoder to detect and remove table entries which have become less frequently used to effect the automatic maintaining the optimal tables.

in the lower levels of the host protocol there is almost always, powerful channel encoding (and as a side effect a robust CRC) is provided by the physical modulation and data link layers or methods to re-send the data in the case of failure. Refer to the host media of interest for specific details. While these message encoding choices are good for the consumer's receiving device, and allow us to place more data on the channel, it does place a somewhat unequal burden at the compression (or encoding) end where presumably computing power and mass storage are readily available. It becomes the job of the ATIS encoder device to recognize patterns of data as they occur and to create the tables then used to efficiently represent these patterns. The basic processing load of encoding can be handled by a desktop grade computer. A very significant aspect of this process is that ALL tables used in this standard are automatically derived and updated requiring no manual assignment or coordination.

As a simple example of this process, consider the names of street segments. Everybody has a "Main Street" in their own hometown. One simple expansion technique is to find an unused character (say one of the non-printing characters in ASCII) and replace all 11 characters with this one symbol. The receiver - decoder device needs to learn to replace this symbol when it occurs with the original 11 characters. What would be more useful is a table of such common phrases, perhaps constituted to fit the local vocabulary needed. Expanding on this a bit further, one can 'throw an exception token' by placing a special character in the flow which is interpreted to mean 'the next character is a table lookup' and then process it accordingly. Often the need for the exception tokens can be removed by context (for example most segments have a post-name such as Street, Ave, Hwy, Blvd, etc.) so one can devise a schema where the last character is always an encoded token.

As is common in many Latin language approaches, the document uses the 8th bit of the octet of ASCII to signify an exception. The value range of 0x81 to 0xFE is then used for expansion based on tables, which are fixed and national in scope. Two types of tables are nested together and used in this document, a double character symbols expansion, and a name string table where the specific table and the way the string is expanded can be selected. The letter pairs are standardized and remains the same in all embodiments. The tables vary with local variation providing additional entries which a standardized National Table does not contain. All tables are automatically generated by the encoder side logic so that no human intervention or coordination is required.

The first 255 entries in the National Table are composed of common pre and post name abbreviations, while the remainder is composed of frequently occurring names. The two letter pairs and the table selection tokens all occupy the non-printing portion of ASCII in the range above 0x7F. Values from 0x80 to 0xEB are processed as two letter expansions. The exception values of 0xEC to 0xFE are used to signify that the next two octets will form an index value into the second and larger table for a further expansion. The range of 0x01-0xFF has been chosen to allow a single octet index into common extensions (St, Rd, etc.). Note again that the exception token need not be thrown when the context is well known.¹⁹ The selection of which table is used is determined by which exception token is used.²⁰

Judicious choices of table entries can result in tremendous average savings. There are a number of well known routines to look over streams of data and automatically generate the tables with the maximum saving. Of course there are always exceptions to the rule such as the low probability of finding a street named "Alameda de Las Pulgas"²¹ in Boston, so as a rule such tables are best computed for the local datum and then issued as an augment to the National Tables. Refer to Sections 10 and 11 for examples.

19. One further variation, a 'toss back' value of all zeros is defined such that in a structure where a table encoded name index is expected but is not available (a very rare event) a table index of 0x0000 is used to indicate that subsequent tokens are an in-line expansion value.

20. The range of eight values is used to index into four different tables. The presence of the even bit is used to indicate that the table entry should be reconstructed with an additional space (" ") character after it. For example, the sequence 0x82 0xZZ 0xZZ would refer to the national names table entry 0xZZZZ which might contain the characters "Main" which would be returned as "Main" by the process.

21. This is a common street name found in California which translates into "the Avenue of the fleas" and runs alongside its better-heeled relative "El Camino Real" or "The King's Road" built by the early Spanish settlers of the area.

The encoding used for strings of text (typically names but not limited to this) allows for four tables to be used, each with its own four local variants of processing. These tables are coded as follows in Table 7:

TABLE 7—TYPES OF TABLES

Table Value Range	Type	Use
0xEC ~ 0xEF	National Table	Static words used Nationally
0xF0 ~ 0xF3	Numbers Table	Static Table of Numeric Values
0xF4 ~ 0xF7	Local Table	Semi-Static words used Locally
0xF8 ~ 0xFB	Temporary Table	Dynamic words used Locally
0xFC ~ 0xFE	Reserved	

The table character is considered an exception value, in that it combines the exception token (values above 0x7F) and specifies which specific table is to be used to process the immediately following 16-bits in the stream which represents the entry into that table. Recall also that the Local Table can vary with the stream in question. In a stream with something other than ATIS text, (say a web page document), the table would be loaded with frequently occurring words for that stream (e.g., http, www.com, etc.).

Each table has four variants set to control the expansion applied to each entry in that table which is encoded in the last 2-bits of the table number itself. These are no additional expansion, adding a final single space character, capitalizing the first character, and finally capitalizing the first character and adding a final space. Observe that capitalization within the name must be pre-established by the entry. Table 8 shows these variants in practice.

TABLE 8—EXAMPLE OF TABLE CONTROL EXPANSION BITS

Table Bits	Meaning	Example 1	Example 2	Example 3
00	No Space, No Cap	<main>	<montgomery>	<elSid>
01	Add Space, No Cap	<main >	<montgomery >	<elSid >
10	No Space, Make Cap	<Main>	<Montgomery>	<ElSid>
11	Add Space, Make Cap	<Main >	<Montgomery >	<ElSid >

One table is solely reserved for expressing numbers and is computed in the receiving device rather than ever being transmitted over the media. This table is considered national and static. The entries in this table are comprised of the numbers from 0 to 65535. If the string value is a number expressed in alpha characters ("Ten" rather than "10") it should be represented in one of the other tables. If it is desired to have a space between the number and the post string, a space should be added while encoding or the extension should be part of the name post string as conforming to local practice. If a value over 65535 needs to be expressed it should be composed of discrete characters (and could be added to the Local Table if conditions warrant). The capitalization bit is modified in this case to indicate adding an ordinal post name (st, nd, rd, th, as needed) and shown in Table 9. As this table is never transmitted, but is fully definable here, responsibility for generating it lies with the equipment manufacturer.

TABLE 9—EXAMPLE OF NUMERIC TABLE CONTROL EXPANSION BITS

Table Bits	Returned Values	Returned Values	Returned Values	Returned Values	Returned Values
00	<1>	<111>	<2>	<3>	<123>
01	<1 >	<111 >	<2 >	<3 >	<123 >
10	<1st>	<111th>	<2nd>	<3rd>	<123th>
11	<1st >	<111 th >	<2nd >	<3rd >	<123th >

Thus, it can be seen that processing a table consists of the following steps which may be implemented in a variety of common programming ways.

Detect if next character in the stream represents a table token
Extract index value from stream
Determine which table is to be used, passing index
Index into table returning string
String expanded from Mod-ASCII
String modified as per variant flags
Receive returned string

A similar method is used for encoding latitude and longitude. Observe that the upper significant bits of all such location data for a given metropolitan area are fixed and do not change. Thus one can devise a simple method to transmit the significant bits only once and then repeat the lower order bits as needed. One can further observe that most street segments are in fact fairly short (under 1/4 of a mile in fact) so one can describe most street segment topologies efficiently by giving its starting point and an offset. Methods which use this concept are presented in the next section.

A review of a simple message being compressed is useful to explore the concepts and benefits of this process. If we consider a segment encoding according to the formats being developed by NTCIP in the External Interfaces effort we have the following two key messages.

The first is the periodically issued update of the link data. For the purpose of illustration we have selected a message which combined the data for all lanes in the link into a single set of numbers. Data from the TMC typically is presented in a *lane by lane* estimate format.

Current-link-state ::= SEQUENCE

{		
link-id-number	IA5String	(SIZE(1..256)),
link-delay	INTEGER	(0..12000),
link-travel-time	INTEGER	(0..10800),
link-volume	INTEGER	(1..100000),
link-speed	INTEGER	(0..300),
link-density	INTEGER	(0..2000),
link-occupancy-percent	INTEGER	(0..100)
}		

The second contains the information about the link and is uniquely defined by the issuing agency (note that it contains the LL data and many other relevant aspects of the link).

Link-description ::= SEQUENCE

{		
link-id-number	IA5String	(SIZE(1..256)),
link-begin-node-latitude	INTEGER	(-90000000..90000000),
link-begin-node-longitude	INTEGER	(-180000000..180000000),
link-begin-node-id	IA5String	(SIZE(1..256)),
link-end-node-id	IA5String	(SIZE(1..256)),
link-end-node-latitude	INTEGER	(-90000000..90000000),
link-end-node-longitude	INTEGER	(-180000000..180000000),
link-direction	ENUMERATED	
{ -- cut from this example }		
link-direction-other-information	IA5String	(SIZE(1..256)),
link-length	INTEGER	(0..160000),
link-capacity	INTEGER	(0..300000),
link-speed-limit	INTEGER	(0..300),
link-speed-limit-truck	INTEGER	(0..300),
link-lanes-minimum-number	INTEGER	(0..50),

```

link-shoulder-width-right      INTEGER      (0..999),
link-shoulder-width-left      INTEGER      (0..999),
link-median-type               ENUMERATED
    { -- cut from this example },
link-median-type-other-information IA5String  (SIZE(1..256)) OPTIONAL,
link-pavement-type             ENUMERATED
    { -- cut from this example },
link-pavement-type-other-information IA5String (SIZE(1..256)) OPTIONAL,
link-restriction-axle-count     INTEGER      (0..20),
link-restriction-height         INTEGER      (0..2000),
link-restriction-length         INTEGER      (0..6000),
link-restriction-weight         INTEGER      (0..50000),
link-restriction-width          INTEGER      (0..2000),
link-oversaturated-threshold    INTEGER      (0..100)
}

```

Reforming this to emphasize only the composite (all lanes) flow rate for the segment as a whole, dropping the additional cross street data, and presuming that there is some type of incident which we are interested in, (and that there is therefore an index and textual explanation to use) we have the following data elements which we wish to send to the community of receivers:

Street Name and Extension
 Starting Cross-Street Name and Extension
 Ending Cross-Street Name and Extension
 Start Lat, Start Long
 Stop Lat, Stop Long
 Current Link Travel Time (here we will presume a simple speed measurement)
 Incident Model Number and Information Text

Let us consider a typical actual freeway segment selected at random from the greater Los Angeles area county and examine the resulting encoding. For simplicity let us drop the cross street names fields.

Pacific Coast Highway	
-118.584804 34.038728	
-118.585412 34.038776	(which is 640-bits long)
30 MPH	
#123 Dog in roadway	

If the geometric values are replaced with the standard 32-bit representation and the other values are similarly replaced (note that the link speed is a 4-bit value²², all others are various octets in length) then we obtain:

Pacific Coast Highway	
0xF8EE8A1C 0x20763C8	
0xF8EE87BC 0x20763F8	(which is 428-bits long)
0x0110b	
0xF123 Dog in roadway	

Converting the street name field, the street extension and the incident phrase into encoded values we then obtain (we are using values of all ones to represent indexes into tables which would be computed in an actual transmission):

22. Five bits allowing 32 values effectively specifying flow rates to a precision of about 3%.

0xFFFF	0xFF	
0xF8EE8A1C	0x20763C8	
0xF8EE87BC	0x20763F8	(which is 188-bits long)
0x0110b		
0xF123	0xFF	

Converting the pair of 32-bit geometric values in the Grid format and adding a value for the Grid-tile index we obtain (the tile is an octet, each point value is 14-bits):

0xFFFF0xFF		- the name index & extension index	
0x1DA00x3B1E		- the point	
0x1D8D	0x3B1F	- the point	(which is 124-bits long)
0xFF	0x0110b	- the tile & the flow rate	
0xF123	0xFF	- the incident index & the text phrase	

We can also safely drop the name and incident text fields as these are fully expandable by indexing off the various tables. The Grid index is transmitted once for all messages in the message frame context so it need not be retransmitted here. The map, which has a unique entry with this set of offsets provides an index to the same name table entry. The incident model is indexed in a similar fashion to the full text explanation. This leaves us with:

0x1DA0	0x3B1E	- the point	
0x1D8D	0x3B1F	- the point	(which is 76-bits long)
0x0110b	0xF123	- the flow rate & incident index	

One further reduction is the observation that most street segments can be fully characterized by using the delta lat/long method rather than needing the full span of the tile in the second set of terms. This leaves us with the following final data which is what is transmitted:

0x1DA0	0x3B1E	- the point	
0x12	0x01	- the delta	(which is 48-bits long)
0x0110b		- the flow rate	

This is basically what is transmitted to update the current link impedance for this segment. Packed into a ≈ 220 Byte message frame, we are able to transmit about 25 such messages per frame or about 100 such messages per second.²³ This represents a compression of about 13x over the original message.

5.3 Tiles and Grids—Efficient Representations of Latitude and Longitude Values—The ATIS messages sent using this document employ the Grid Profile to represent latitude and longitude. The Grid Profile was developed for use in SAE J2369 to maximize the use of limited bandwidth mediums while still allowing a nationally interchangeable methodology among different map vendors, and data sources and still be scaleable for growth. Bit level documentation of the Grid Profile and of other profiles used in ITS, and in particular in vehicular applications, can be found in the Location Reference Message Set (LRMS) documentation. Explanations of implementation guidelines are presented here. The Grid Profile can be readily expanded back to the other more verbose profiles as the application requires. In general, manipulation of the data in the compressed format is unusual, being limited mostly to methods for filtering and sorting to reject data not of current interest to the end user²⁴. The name grid comes from breaking the datum of latitude and longitude down into tiles or Grids, which are then encoded for transmission in this profile.

23. As a design goal during development, we have used a 5000 segment model which must be updated every 3 min as our benchmark. The above rate would need the channel for about 50 s, therefore requiring about 28% of the total channel capacity on an FM SubCarrier host. FYI: Values above $\sim 40\%$ are thought to be prohibitive in terms of subcarrier business model costs.

24. Keep in mind that the area of coverage for a given media broadcast may exceed the area where a given end user may be traveling, hence the desire to filter data which might only be an interest to travelers "on the other side of town".

An overview is presented here to understanding the process of translating *into* and *out of* the Grid Profile DEs. At the transmission end full 32-bit lat/long values are extracted from the other profiles of the LRMS and are prepared for transmission by a two-step encoding process which is reversed in the receiving device. As with all SAE J2369 messages, an initial “context” is established which sets various default values (including the specific Grid) and then the data in question (which has been reduced to an offset from this value - thus requiring less dynamic range to represent) is transmitted over the media. Collections of similar geographic points (which occurs often in ATIS broadcast applications) allow the context to be sent once per message frame followed by multiple data points. Such a flow of “context” followed by sets of data is referred to as a “stream” and is portrayed in Figure 7.

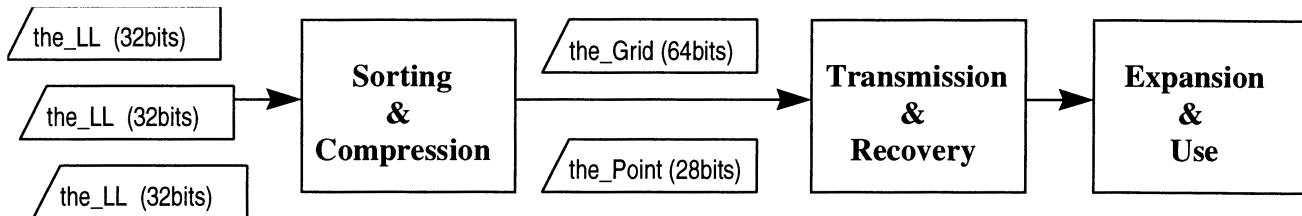


FIGURE 7—CONCEPTUAL FLOW OF DATA USING THE GRID PROFILE

Streams are a key concept in the SAE J2369 messages. It is sufficient for LRMS purposes to understand that there are different types of streams. For example, there are streams of 32-bit lat/long pairs, of points, and of segments in the efficient encoding format of this profile, and of Grids which are required to interpret these encoding. Streams may also contain encoded data such as street names and other useful tables. A network directory layer allows streams to be easily ignored or filtered once recovered to save processing bandwidth (because it contains the sizing and content data described in the four level flow schema described earlier). All streams are framed into a message of from 20 to 512 bytes as supported by the host media. The key point is that within each message frame the context is set once in the beginning of the message. The specific details of context setting by a prior message element are contained in the section on the message set frame message protocol.

In the receiving device, the packet is received from the token data stream, the context is recovered and used to set default values, then sets of encoded data are processed as needed. Typically, the Grid Profile data is expanded into a less efficient but more generally useful format for internal storage, such as pairs of 32-bit values²⁵. This process involves using data from the collection of pre-defined Grids, called “tile objects” which form the most significant part of the lat/long and adding to them the “point object” to reconstitute a complete 32-bit value. The context data contains a value which is used to index into the array of tile objects. The tile objects are obtained from a stream of tile data (which is periodically transmitted in the table data stream of the same schema). In actual processing, this requires a shift and add operation for every point which needs to be expanded into a 32-bit representation for storage.

With this background, we will describe the elements and their bit structures followed by the object definitions in greater detail. We define the class tree of the five objects used in the Grid Profile. Specific bit representations can be found in the DEs sections. A simple beginner’s example of translation into a Grid Profile can be found in the previous section on data compression.

First we define the basic “the_LL” data element object, a 32-bit quantity which is defined as being encoded with the 29th bit as exactly 1 microdegree (= 0.000001). The LSB of objects of this type are then 1/8 microdegree in value. This standard data element is used across the ITS community. This definition provides for an LSB of under an inch while effectively allowing a global range of encoding in a 4 octet value suitable for memory storage. For the wireless community, it has been determined that the needs for precision (not accuracy which

25. Refer to SAE J2354 for examples of navigation messages containing ATIS data.

is of course less) are for a level of 32 microdegrees. This corresponds to about 11.68 ft at the equator, and somewhat less in North America. Therefore, this profile was targeted to support this level of granularity very efficiently.

If a greater precision is required, then a full 32-bit value can be transmitted at the expense of bandwidth, or the features of the “zoom-scale” method of the tile objects can be employed. With the exception of the 32-bit “the_LL” data element, all other methods of this profile (at the nominal zoom) resolve to increments of 32 microdegrees.

Tiles are selected for use by establishing the position set for “the_Grid” which is then entered into a table and periodically transmitted among the streams. Barring any other considerations, the position of Grids should be chosen so that adjacent tiles overlap by a 50% margin and align themselves on a 0.262144 degree boundary (the 10th bit, approximately 1/2 degrees). Thus a generic layout would result in a Grid of tiles overlapping by 0.262144 degrees and each 0.524288 degree on a side and on a boundary of 0.262144 degrees. This corresponds to the nominal zoom of zero. Regardless of this overlap, segments should be expressed in the same single tile each time to aid in receiver processing. In the very unlikely event of a single roadway segment being longer than 36 miles and extending beyond a single tile, then a forced node must be added to maintain the encoding rules.

The tiles need not be placed on this strict boundary to work. Receiving devices are required to be able to interpret any valid Grid value and to be able to add it to points and deltas to determine full 32-bit values. The reason for setting a non-standard tile (with the standard zoom of zero) would typically be a local need to cover a disjoint land mass such as caused by a water body. Under such conditions, the Grid values should be chosen to achieve the previous overlaps in so far as practical, however the Grid values for the tiles can be set to a precision of 0.57 miles (0.008 degrees, the precision allowed by the 16th bit of the tile datum). This effectively allows the local agencies (as a part of setting up their shared datum rules) to place the tile anywhere they desire. As a recommended practice, adjacent tiles should still overlap by 0.262144 degrees unless conditions merit. Tiles for exchanging data on the same scale (e.g., the local tiles) should use the same zoom scale, nominally the zero zoom.

Finally, a single tile should be set at a large zoom and placed in the table which covers the entire metro area including all outlying areas for which local data could be sent. The national tiles can be used for this purpose if a resolution of ~747 feet is sufficient for local needs.

The tiles should be ordered in the table area starting with the national tiles, then the major local tile (if present) then local tiles in order from most south-eastern to most north-western. Tiles with smaller zooms are placed after these tiles with larger zooms. Vanity tiles are placed last in order. The ordering rule is to place larger first followed by smaller, closer to the primer meridian and equator then farther, and finally having vanity tiles appearing last in the list. Receivers not processing the available vanity maps are free to discard these last entries.

The range covered by valid *the_LL* elements and combinations of all other elements is ± 180 for longitude and ± 90 degrees for latitude. The range covered by *the_Point* is unsigned and therefore taken as positive. It moves further eastward or northward with increasing values. In North America, which grossly runs from -60 to -120 degrees in longitude and is in positive latitude, the values selected for tiles with *the_Grid* are in the lower western corner of the areas which they cover. The value of *the_Point* added to *the_Grid* then places the coordinate point of interest further north and to the east. It can be observed that the use of the signed *the_delta* in conjunction with a *the_Point* value of zero would result in a coordinate position which is outside of this range. This will not occur in practice due the restriction that *the_Point* is only employed for short segments and the chosen 50% overlaps between adjacent tiles. See Table 10.

TABLE 10—THE STANDARD 32-BIT LAT/LONG VALUES BY BIT VALUE

Bit	Distance at Equator		Bit 29 = 1 micro degree
1	24 889.78	mi	Sign
2	9 279.58	mi	134.2177280
3	4 639.79	mi	67.10886400
4	2 319.90	mi	33.55443200
5	1 159.95	mi	16.77721600
6	579.97	mi	8.38860800
7	289.99	mi	4.19430400
8	144.99	mi	2.09715200
9	72.50	mi	1.04857600
10	36.25	mi	0.52428800
11	18.12	mi	0.26214400
12	9.06	mi	0.13107200
13	4.53	mi	0.06553600
14	2.27	mi	0.03276800
15	1.13	mi	0.01638400
16	0.57	mi	0.00819200
17	0.28	mi	0.00409600
18	747.62	ft	0.00204800
19	373.81	ft	0.00102400
20	186.91	ft	0.00051200
21	93.45	ft	0.00025600
22	46.73	ft	0.00012800
23	23.36	ft	0.00006400
24	11.68	ft	0.00003200
25	5.84	ft	0.00001600
26	2.92	ft	0.00000800
27	1.46	ft	0.00000400
28	8.761	in	0.00000200
29	4.381	in	0.000001000
30	2.190	in	0.000000500
31	1.095	in	0.000000250
32	0.548	in	0.000000125

The concept of zoom is used to shift the scaling of the map by powers of two up or down. A value of zero is considered nominal and indicates the values shown in Table 11 for each bit. This is the setting used for the vast majority of applications. Zooms can range from +6 to -5 in value.

A “large” zoom is a positive value which is taken to shift the application of *the_Point* and *the_Delta* UPWARD in magnitude with respect to mathematical operations with *the_Grid* (which is not changed and hence keeps the same resolution of the 16th bit = 8192 microdegrees). Said another way, the zoom provides a shift up or down function to point values operating on this tile. This has the effect of doubling the range of *the_Point* but halving the resolution for each step of zoom. For example, the National Table of tiles uses a Grid set to -135 degrees longitude and +15 degrees latitude which covers a vast area of the Pacific Ocean (waste) as well as all of California up to nearly Portland and eastward over the entire southwest to Denver. This one single tile (with a zoom of +6) is used to report weather conditions and other large scale events. The corresponding loss of resolution is such that the LSB of the *the_Point* is now changed from 32 microdegrees (<12ft) to 2048 microdegrees (~750 ft).

TABLE 11—SCALED VALUES FOR LSB OF THE_POINT WITH ZOOM SETTINGS

Zoom	Value of LSB of the_Point	
6	2048 microdegrees	National Entries ~750 ft
5	1024 microdegrees	
4	512 microdegrees	
3	256 microdegrees	
2	128 microdegrees	
1	64 microdegrees	Nominal ~11.69 ft
0	32 microdegrees	
-1	16 microdegrees	
-2	8 microdegrees	
-3	4 microdegrees	
-4	2 microdegrees	Smallest Scale ~ 4.38 in
-5	1 microdegrees	

Zoom can also be used to reduce the range of the tile covered by a specific Grid. This can be used to handle various small scale maps (including vanity maps) as might be found in a theme park or a hotel complex. Repeating the previous, we have: A “small” or negative zoom is a negative value which is taken to shift the application of *the_Point* and *the_Delta* DOWNWARD in magnitude with respect to mathematical operations with *the_Grid* (which is not changed and hence keeps the same resolution of the 16th bit = 8192 microdegrees). This has the effect of halving the range of *the_Point* but doubling the resolution for each step of zoom.

Finally, one or more vanity maps may exist in the system and will have one or more defined *the_Grid* entries in the table. A vanity map is used to convey a map which is not strictly correct with respect to ground topological features. A transit map is a typical example. An amusement park or hotel complex is another. In both cases these maps can use the Grid entries (complete with zoom). The relationship between the map and the “real world” of the other tile entries is not specified here (if it exists at all) but is specified in the map message for the map. Note that often these maps do not name the segments, rather they often name the nodes (a train schedule calls out names of each station, not the track between).

5.4 Traffic Models—Shared Models of Link Impedance Modeling—As first described in the overview, this document uses the concept of shared parametric models to convey information about link impedances. This section further elaborates on the process used and the desired goals for the models. The model process is normally applied to traffic link flow impedance, however, it can be applied to any time varying value (for example, the rate of snowfall in a weather related information stream). The practitioner must have previously developed suitable predictive data which he desires to model which is beyond the scope of this document.

The models consist of several simple shapes representing common flow occurrences. See the Appendix for a complete list and precise mathematical shape definitions. In each instance of a model, there is also values for amplitude, start time, and duration. These are then reused by attaching them to specific segments and scaling them as needed. Each model is of “unity” height and width, this is scaled to suit the need based on the supplied parametric values for time duration (width) and amplitude (height).

The models may be used in a lumped and combined manner or a new model may replace prior models attached or related to the same segment or incident. Said in another way, by setting a selected bit, the model can have the property of superposition or it can replace a prior model. This allows both very efficient encoding of typical events needed to support forward traveling planning²⁶ as well as being a simple way to handle incidents which propagate all over the network. By setting the time values approximately, the basic models can be concatenated to develop more complex shapes.

The models are attached to the segment of interest by a two-step process as shown as follows in Figure 8. The message which refers to the segment (with LRMS means) contains a data element which provides an index to a table incident number. The table in turn points to one or more actual models in the incident model table and also can refer to textual description data and other items not shown in Figure 9. The table provides the scaling data for the model as used by the specific segment.

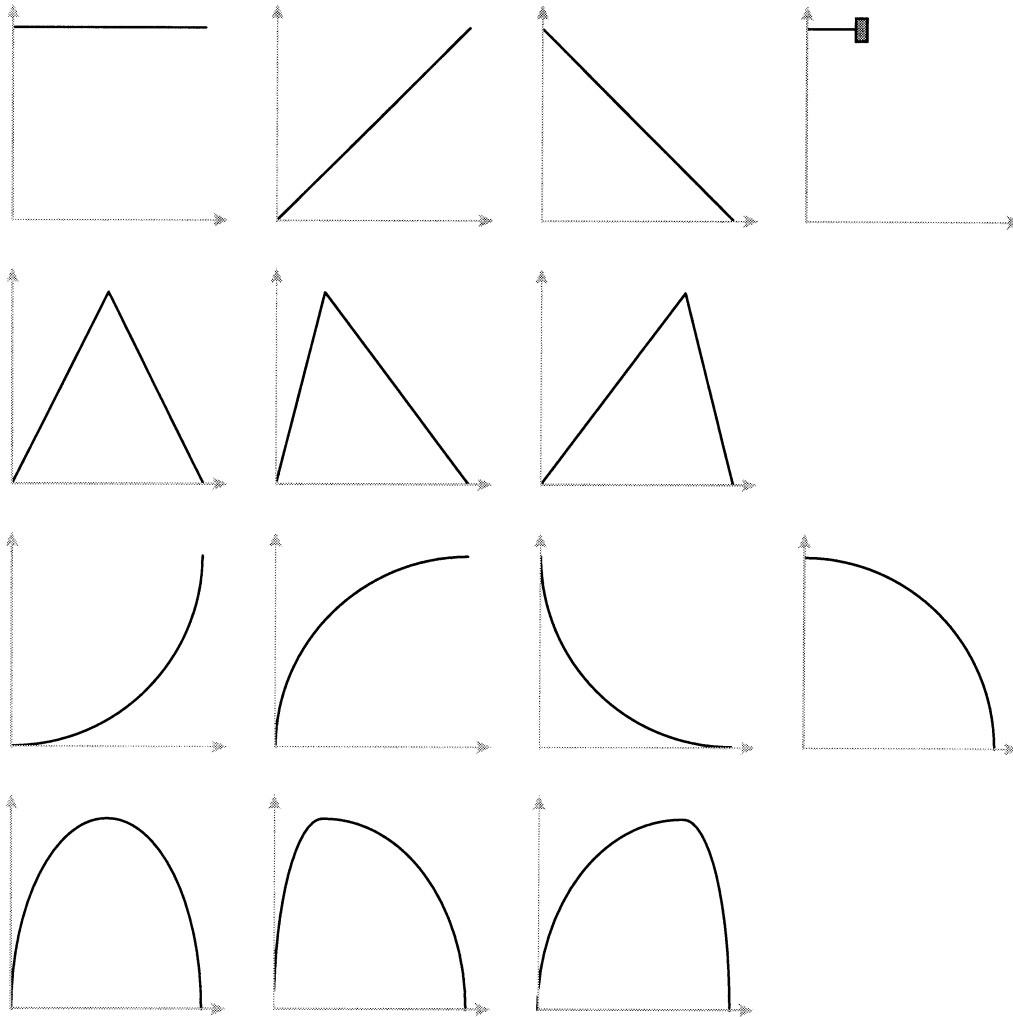


FIGURE 8—BASIC MODEL SHAPES

26. One of the key goals of this document is to advance the art so that a traveler, when not in communication with the host ATIS link, can still determine what the "normal" flow expectations will be on supported links. In end user devices, this manifests itself in two primary functional areas. First, to be able to transverse the links for models to predict what the *segment by segment* travel times will be, *at the predicted time the traveler reaches them*. Second, to plan future travel by considering the expected flows based on the self contained models previously received.

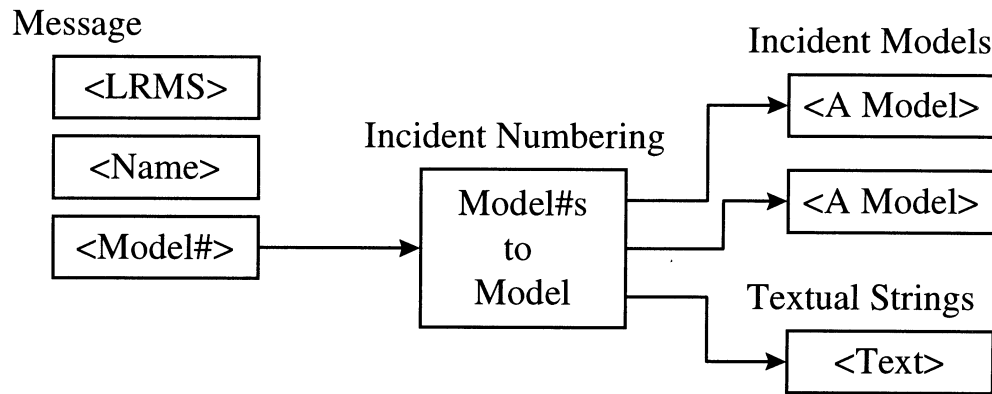


FIGURE 9—RELATIONSHIP OF INCIDENT FLOW MODEL MESSAGE TO ACTUAL MODEL ENTRIES

Each model entry consists of one or more 40-bit fields containing the parametric values of that model instance. Refer to the content header for details of message packaging. Refer to the messages and data elements for bit level details and binding between models. The models contain the following data elements (see Tables 12 and 13):

TABLE 12—CONTENTS OF A MODEL ENTRY

Element	Commentary	Size
Day of Week Code	Each day of week applicable	8-bits
Start Time, Evoke Time	A range of 8.5 days, a precision of 3 min	12-bits
Length Time, Persistence	A useful range of > 4 h, a precision of 3 min	6-bits
Amplitude	A range of -15 to +48 (-20% to +100%, in 1.9% steps)	6-bits
Model Shape	A choice of 16 shapes	4-bits
Event Type	A choice of 16 types (pre-planned, etc..)	4-bits
		40-bits total

When used in-line for *segment plus incident* messages, some of these elements are not transmitted, others such as start time can be shortened from context. In this use, the time from which the model is indexed as provided in the context.²⁷ Messages of this type would not typically be kept by the user device between power down states, while the previous message would if resources allowed.

TABLE 13—CONTENTS OF A MODEL WHEN USED IN-LINE

Element	Commentary	Size
A superposition bit	Token used to effect other models	1-bit
A direction bit	Direction token	1-bit
Start Time, Evoke Time	A range of 4 h, a precision of 3 min	6-bits
Length Time, Persistence	A useful range of > 4 h, a precision of 3 min	6-bits
Amplitude	A range of -15 to +48 (-20% to +100%, in 1.9% steps)	6-bits
Model Shape	A choice of 16 shapes	4-bits
		24-bits total

27. An obscure and counter intuitive aspect of encoding the time from which short item fields are offset is that one does not want to update this time mark more often than required for the benefit of the receiver population. The receivers, who are charged with translating the messages into internal user memory, can compare the frame CRC to determine when a set of models has been seen before and thereby requires no further processing when repeated in transmission. If the model's encoded time is shifted along with the transmission time with each repeated frame, the receiver must process the entire frame to update bits and hence use processing resources. If all that has changed is the relative time stamp mark and not the model data itself, this is wasteful. Thus, the model time should only be changed periodically. With a range of multiple hours, most incidents will stale out without requiring an update. When an update occurs, all receivers will be burdened to process the message again.

Under in-line conditions, the data is sent (often with other data) in the message as shown as follows. Here no reuse between segments can occur, the data must be repeated for each use. As such, this is not the recommended practice for normal congestion conditions but should be used for recently occurring incidents which have not yet been added to the table or where the predictive nature of the roadway interruption is not yet known with any degree of certainty.

The models are by design fairly crude and allow a modeling implementation precision on the order of 3%. With current state-of-the-art daily traffic models, a 10% error in prediction is considered to be very good and perhaps 25% being typical. State-of-the-art for short term use is around 5%. Due to the gross nature of such predictions, this document uses model amplitudes to a 2⁶-bit value which provides resolution of ~2% sufficient for our needs. The value range is offset such that it covers a range of -15 to +48 (positive being a reduction of flow in this context).²⁸ The offset range allows the model to be used when flows are better than expected or the prior model would indicate. See Figure 10.

Message

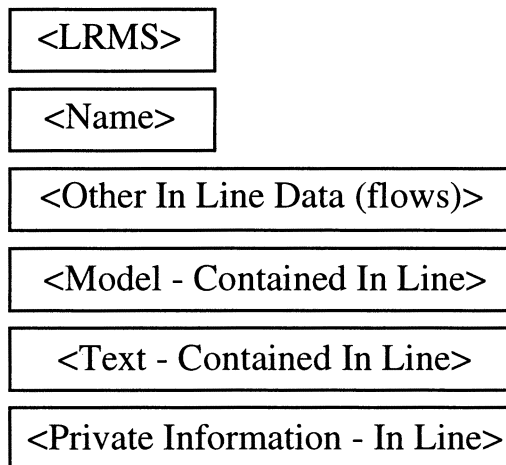


FIGURE 10—MODEL USE WHEN TRANSMITTED IN-LINE WITH A MESSAGE

In practice, when an incident occurs and superposition of models is in use, new models can be added to reflect an incident on top of the prior "normal" models. Such models can be selected to have an duration commensurate with the incident clearance times. In the figure which follows, the gray is the actual rate, the models reflect the instantaneous issued updates during the event (short lines in upper right) and finally the new additive model once the incident has stabilized and is in the clean up mode (lower right) relaxing back to normal flow rates. See Figure 11.

28. In this document, we use the concept of positive value to imply the reduction from free flow.

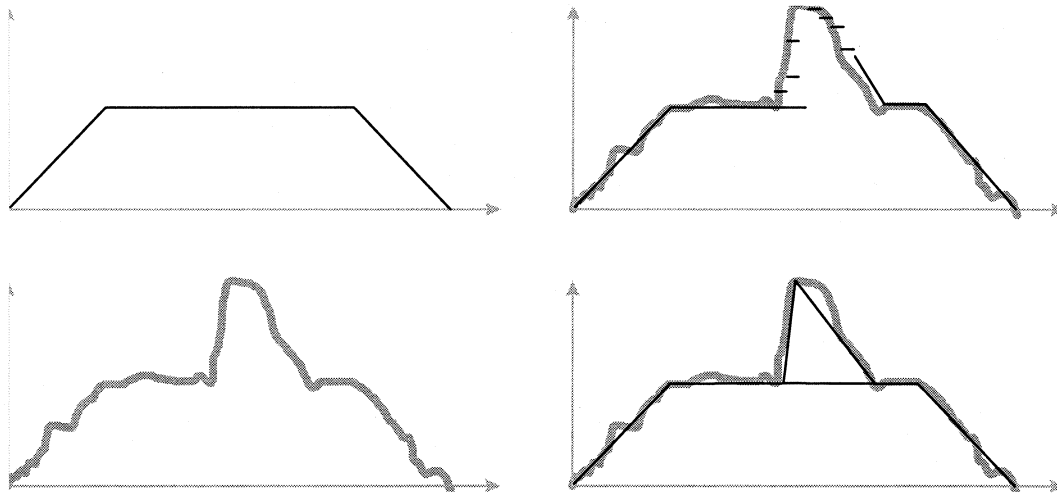


FIGURE 11—MODELS WITH AN ADDED ELEMENT DUE TO AN INCIDENT

Another event, which commonly occurs, is when flow rates are better than anticipated by the model.²⁹ Under these conditions, a model with a negative rate can be issued at the time the event is understood to adjust the summary flow to reflect conditions. In Figure 12, an event has allowed the actual flows (in gray on bottom) to be better than the model predicted. Two different corrective actions could occur and are shown. In the right top figure, a model with a negative flow is issued. In the figure at bottom right, a new model has replaced the old original predictive data.

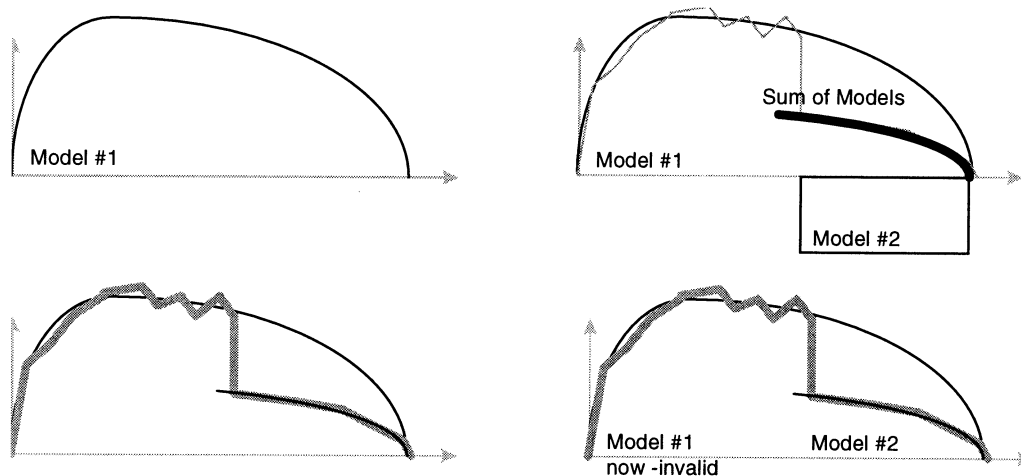


FIGURE 12—COMPARISON OF A MODELS WITH A NEGATIVE FLOW RATE

29. Before the reader thinks this is a good thing, such events usually occur just past the incident event. Most of us seem to be in the congestion on the upstream side.

In all of these scenarios, the ISP or data source can issue either corrective models, or switch to non-additive models when conditions warrant. The data source can also issue rapid update by update rates without any predictive value. This choice can be made either globally or on an *incident by incident* basis.

The time element of the models allows for a variable precision for both start times and for duration times. Some bit settings are also reserved for conditions such as "indefinite" or until further notice. A range of eight days allow pre-planned events to be sent ahead of time. In cases where data is attached to the transmission of a specific segment, the range for time is decreased. In both of these events, time is referenced to a time marker in the context which is slowly updated. Receiving devices will likely translate this into whatever concept of time they implement.

Note that while the time and amplitude values are scaled to be fairly crude, the resulting value taken from the curve is not. In use it is determined to whatever value of accuracy the implementers choose to create. Values of ~0.4% error occur with byte long results.

The models are used for all types of incidents and do not in themselves care what the root cause of the incident event is. This information is conveyed in the *event type* tokens and can be further investigated by comparisons in the other incident flows. These types of tokens are general in nature and do not replace the RDS derived textual descriptions.

If no predictive model information is available, then the set rate model (see 12.4 Model #3) can be used to create a message whose meaning is "the flow at present is X" - this allows localities with no predictive abilities to still use the standard.³⁰ For those with information regarding daily patterns, the model allows multiple models segments to be created and the applicable usage (a bit field for each day of the week), the most common example of this is a bi-daily routine commuter flow. For those with short-term predictive model data ("we believe flows will resume over the next 20 minutes"), the model allows a time stamp from the moment issued to be used. This is typically used in the "relaxation" stage of an unplanned incident.

The use of the 8-bit field for days of the week was chosen over the common alternative, a list of: weekend, mid-week, Monday, Friday, and Holiday. In this model, the 8th bit is used for Holidays, and no bits are used if the model refers to a specific, one-time non-reoccurring event (such as a major event's flows).

Because the models are parametric, each is used by attaching a link and providing scaling amplitude for that link. Each model provides the time varying part. They may be made up of one or more models lumped together. Many links will use the same model to reuse processing and memory resources each scaling to the amplitude, which makes sense for the units they wish. Typically several connected links forming a commute route might make reference the same model with decreasing offsets as the "bottleneck" dissipates. In Figure 13, we show the flow rates for a set of segments [A to G] where a bottleneck in the central segments [at E] contributes to up-stream and down-stream flow reduction in a repeating daily commute example. This example uses one model instance (40-bits) reused by each of the seven segments.

These models are efficient and handle the anticipated needs for the traveler to obtain current data, plan future routes, and automatically use the data in dynamic route guidance systems. While it is a good first step and presents a standard upon which to grow, it does not address some other related issues.

There are several shortcomings in the modeling structure selected for use at this time which should be reviewed. We have adopted here a *flow rate* model, some practitioners prefer a link *travel time* model. While the two are related they are not just a simple conversation of units. As a gross approximation, one can multiply the flow at the time the link will be transversed by the length of the link itself. In surface streets areas with sizable intersection signaling times, this approach breaks down.

30. As pointed out in Section Four, this can be fairly inefficient in terms of bandwidth due to the constant need to repeat the message for new users, and of course it does not allow the user to predict anything.

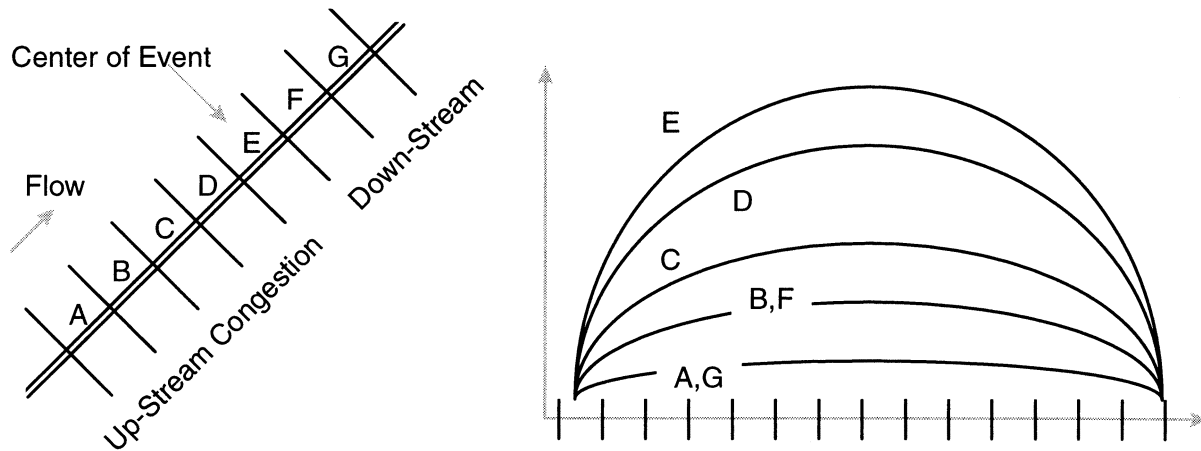


FIGURE 13—SIMPLE REUSE USE OF MODELS IN A TRAFFIC FLOW EXAMPLE

There is no provision for modeling a variance or traffic density information. While this tends to only be a problem at low occupancy rates, some consider it useful data. The models adopted here have no provision for accuracy or quality metrics. There are no provisions in the current model to handle link centric flows such as the time to traverse a traffic intersection or a ramp metering event. There are no provisions for a stochastic delay element. While many TMCs are still struggling to model flow at all, these concerns can be ignored for the moment.

The addition of at least a stochastic delay element to represent the bounds of traversing, a traffic intersection will also be reused to estimate bus route next vehicle data on routes where a vehicle is expected periodically³¹, a phase II area for standardization.

5.5 Forward Pointers—Indirection to Other Data Sources—The concept of forward pointer is used to direct the user to additional streams of information about a subject (typically an incident event) which are contained elsewhere. Two types of elsewhere are defined at this point, pointers to other media and locations on the World Wide Web (independent of the access technology used). This table may be expanded to include other technologies at a later time.

The forward pointer format consists of three tokens as follows. An index value used to refer to the entry in the normal table formats as a sparse table type. This is how other messages refer to this pointer. This is followed by a single entry (which can be viewed as part of a string) where the technology is represented in the first byte and the remainder of the address is contained in the string. The string may use table compression and will be expanded by the table indicated in the context values section. The string may also use the national and local tables in the normal way. It is anticipated that a local table would be filled with tokens such as "www.etc..." as local condition require.

In the case of a SubCarrier reference, there are 200 carrier frequencies for FM stations to exist on, each of which may have as many as three SubCarriers, hence an encoding of 10-bits is used.

31. This is the case of "another bus every 15 minutes" as opposed to a fixed asset assignment on the route as is found in other transit schedules or in airline flights such as "flight 123 departs at 12:34."

5.6 Frame Bit Stuffing—Unused “Flea Bits” in the Message Frame—The packing of information into a fixed frame of bytes does not always result in a complete payload fitting into the packet. Under these conditions, the remaining bits are reserved for use by the encoder and do not constitute information of value to the ATIS user. These bits are referred to as “flea bits” and may be used by the encoder to transmit low data rate information in any way that it desires. It should be observed that the beginning of the flea bits can not be determined until the message itself is decoded, hence any use of this information must be capable of decoding the precise length of the payload section in each message.

5.7 Frame Headers—The Concept of Context—A critical concept in the encoding of the message is that of the *context* which is established at the beginning of each message, also called a frame header, and which is defined at length in Section 7. By way of an overview, it suffices to say that the context defines all the default values for the messages which follow in that packet. This includes the presence or absence of various data elements in the messages as well the specific tables used to decode information in the messages.

The essential summary is that these bits are used to describe the formatting of the data which follows. The basic structural choices for each data element are: *present*, *not-present*, and *maybe-present*. In the first two cases, the expectation is defined for the messages. In the *maybe-present* case, there will be a test bit to determine the presence or absence of the data element in question on a *case by case* basis. These definitions are provided in the context section for each type of message. This concept is used by other ATIS practitioners and is described by them with different terms. The term “self describing data” has been popular in the US to refer to this structural construct. In the European community, the term “bit field” has been applied to context flags portion of the frame header.

The key concept is that the information in the frame header is subsequently used to decode and to expand the data contained in the messages which follow.

5.8 Transit Scheduling—Unified Concepts of Multi Modal Scheduling Tables—This is an area for future expansion of the message set in the Phase II effort. Following some of the requirements and design goals for this set of messages are discussed.

Transit is by nature multi-modal and there is a basic need for route tables and adherence to same regardless of the mode of transport involved. The table structures developed here will be employed to relate the schedule and the adherence to it when such data is available. We expect to support multi-model uses including commercial airline schedules with these messages.

Two varieties of schedule support are foreseen: A specific route and resource scheduling model (with a specific bus, train, planes on fixed routes, e.g., flight #123) and rotation schedule types (e.g., a new one every 10 minutes).

There are some issues of what attribute data versus community bandwidth needs to support (wheelchair accessible routes etc.). The ability to self-define messages using the context flag method is foreseen as being able to solve these issues.

Finally transit routes are often displayed to the consumer public using maps which are not strictly correct but which convey a visual routing impression to the end user. The vanity mapping constructs will be used to overly the transit routing data with the rest of the datum in a unified way.

6. Network/Directory Messages

6.1 Overview of Structure and Stream Mapping—The stream of messages which are delivered over the host media, must provide a means to map the location of messages of this document within the host media. Within whatever range of assignment is provided, this document then assigns the message type in the initial byte of the message packet, see Table 14 for these assignments. Note that the entire 8-bit (1 octet) range is reserved for future assignment.

Some host media utilize the concept of a network service provider as a means to allocate useable bandwidth. In such cases a “non-aligned” network ID shall be provided by the assignment authority of the host media such that implementers of this document which do not have such an identity (as may often be the case with local public officials implementing ATIS systems and therefore only casually in the business of telecommunications) can use this ID assignment.

In summary, the first byte of the message is used to differentiate the various ATIS messages within the structure provided by the host media. The entire range of 0 to 255 is reserved for use by this document and its successors. In some media additional mappings may be required but in all cases can be related back to data frame structure. The mapping of message types to this range is provided in Section 7. See Figure 14.

Data Frame, ~20 to 512 Bytes



FIGURE 14—THE PARTS OF A MESSAGE PACKET

6.2 Message and Table Stream Numbering—The mapping of specific table numberings within the message types are provided in Section 7. Within this ATIS document, several key tables are defined. They are treated as simple messages with a common set of delivery formats. This point of view was adopted to promote the reuse of these message formats and compression structures among different applications of the media.

7. Message Frame Headers Elements (Context)—In this section, the *data frame header* elements are defined and described for each type of ATIS data stream. The data frame itself is a multi- byte long message, which is recovered by each receiver based on the assignment of ATIS messages to specific physical streams. The first byte of the frame consists of the message type, which is used in this schema to indicate what type of data is present (e.g., link times, incident text, tables, maps etc... all within the realm of ATIS streams). The next several bytes (four or more) contain the context information in a header. The remaining bits (up to any flea bits) contain useful information referred to as the *payload*.

The data frame header (henceforth referred to as: header) precedes the payload bearing section of every message and provides the context for the subsequent data items in the payload. The context consists of three parts, the context flags, the table of default values, and some data count and sizing information which is placed at the beginning and end of the context. See Figure 15.

Data Frame, ~20 to 512 Bytes



The Context

FIGURE 15—THE PARTS OF A DATA FRAME

The *context flags* section provides the precise *type* definition of data elements used in this frame. The data elements used in each item (in terms of contents not in terms of order) may be defined on a *frame by frame* basis in this way to achieve the best use of bandwidth and of local practice. This type of variable message format is referred to as self-describing by some practitioners. The guiding concept behind this construct is to be able to send only that information which changes, and not to send information which is static across the frame. Most of these flags are set to a value which is either TRUE (datum is present), FALSE (datum is not present) or MAYBE (indicating that a single bit will be inserted just before the datum to determine the presence or absence of the subject datum). For example, if the data is a simple lat/long pair with no name index entry, or vertical data, or other LRMS elements, the flags can be set to represent this simple structure. If the LRMS datum being used may contain a street name index in the items to follow, the MAYBE bit is set. By setting this bit, the decoder knows that it must look for the specific single bit flag on each and every item to determine if there is an entry.³² Context flags can be set by the encoder based on what data is presented to it and what compression it can achieve on each individual frame. Alternatively, on a local level, context flags can be forced to achieve highly inefficient data compression.

The *context table* section provides default values for some of the data elements needed in expansion. For example, the base tile which is used in the grid method is defined here. If the flag indicates that a datum value, which needs a table entry is not present, then that table entry is also not present. For example, if an expansion table context flag for a custom string delimiter is not set, then there will not be an entry for the custom delimiter value in the table section. The table items always occur in the order specified, therefore, by decoding the flags no ambiguity in table entries can occur. To assist in understanding, a letter designation is used in the definition table of each section of the document to help the reader to map context flags to context table entries used.

Lastly are the count data items. There is a 9-bit *item count* containing the number of unique items that are contained in the payload. This is followed in a 7-bit *item byte count* describing the payload area in octets of data. These values are unsigned. The final bit of the final item is contained within the final byte, however, this typically does not fall on a boundary divisible by eight on host media which are not byte oriented. These two elements are placed after the context table data. Finally, there is a CRC-16 value which covers the entire frame (bytes two and onward of context and payload). It is placed as the first two bytes in the context. The purpose of this value is not error detection but rather to easily determine if this frame is a duplicate of a recent prior frame.³³ The order of appearance (and hence the decoding) of data elements in each is predefined. See Figure 16.

Data Frame, ~20 to 512 Bytes

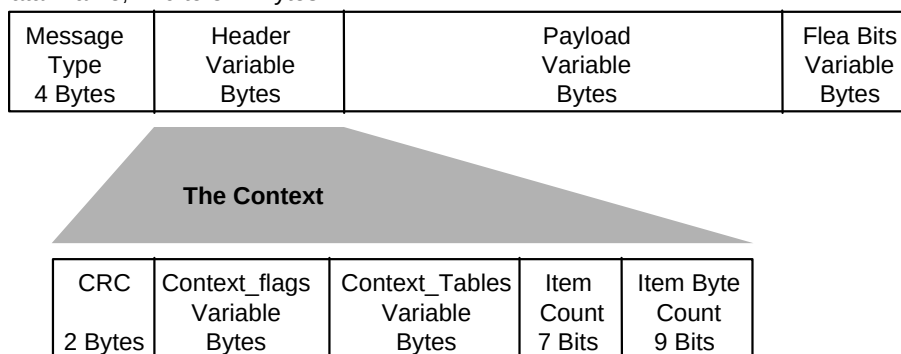


FIGURE 16—DETAILED CONTEXT INFORMATION

32. A reason this would be done is in order to differentiate two segments which might otherwise be ambiguous. Note that if this is done ALL the time, a different context would be set at the encoder. The MAYBE flag is typically used in those cases where a generally more efficient encoding flag setting will not serve the needs and an exception must be made. Note that this costs at least one bit per each use of MAYBE per each item - typically this is much less than repeating superfluous datum element in each payload item, or adding a termination character for strings.

33. If the CRC value is the same as one recently processed, then this frame almost certainly contains no new information and need not be processed again.

In the payload section following the context are the number of items indicated, each of variable length based on their final encoding but are constrained by the byte count provided. Items occupy only the number of bits required and do not adhere to any byte boundary rules. Subsequent items start on the next available bit in the payload.

Data streams not related to ATIS message flows need not adhere to data frame and the payload aspects the protocol. Once within the first four bytes of the data frame, non-ATIS information providers are free to choose encoding in any way which suits their needs. They are also free to use the tables and techniques developed for ATIS, however, they must publish their formatting rules to their user community. Note that while ATIS flows are not encrypted (the first bit indicating this), they do adhere to the community encryption rules. Non-ATIS flows are free to use the encryption or not as they see fit. They are also free to use additional encryption, either additional or in place of the standard forms.

The streams for which a context is defined are as follows in Table 14. Some of these streams are reserved and will be defined in the next phase of standardization. A number of other tables which are not used in ATIS (such as stock reporting or music song titles) are also not standardized in this document but may be of interest to the reader. The streams which are defined in this document are indicated with a check mark.

TABLE 14—LIST OF STREAMS WITH CONTEXTS

Stream Name	Defined	Msg Type Id
• Bin Data	N.A.	0x 04
• Expansion Tables (general names & table use)	√	0x 08
• Forward Referencing Tables	√	0x 0C
• Grid Array Tables	√	0x 10
• Incident Flow Data	√	0x 14
• Incident Modeling Tables	√	0x 18
• Incident Numbering Tables	√	0x 1C
• Map Encoding (general base map use)	√	0x 20
• Private Data		0x 24
• RDS Incident Encoding		0x 28
• Resource Availability (lot availability, ride share)		0x 2C
• Textual Incident Encoding	√	0x 30
• Transits Schedule		0x 34
• Weather Data		0x 38
• Vanity Maps		0x 3C
• Alternative Names Tables	√	0x 40
• Table of Tables	√	0x 44

In the previous numbering, a space of 4 adjacent messages has been reserved for each message type for future expansion of messages related to that type. Conceptually, a sequence of related messages (up to four) would be allowed to progress together in concatenated transmission order and would employ only one context header among the collection of them to achieve higher effective throughput of ~2048 bytes. Similarly for extremely reduced bandwidth hosts, the previous numbers can be reduced or eliminated in transmission along with the context portion of the messages IF at the receiving end a complete message in the SAE J2369 format can be reassembled, effecting a means to send “payload only” messages of ~10 bytes per message. Implementation of such extreme formats is not recommended. All message assignments from 0 to 255 are reserved for use by this document and its successors.

Each of the various streams of data are divided into data frame messages and each data frame has a header with a context associated with it. Specific meaning of bits for each header and for each message are described in turn, except for the count items which are the same for all contexts and are defined as:

Item Count	(0~127) A count of the number of items to follow.
Item Byte Count	(0~511) Count of bytes of payload data to follow. Any bytes left over in the frame may be used for flea bits.
CRC	A CRC ₁₆ value for the entire frame which can be used to determine if this frame has been processed before.

Within the various messages are tables which are required to decode the messages. These tables are referred to by an 8-bit assigned number ID. Some of the ID numberings assigned to these tables are pre-assigned as indicated in Table 15. Others (typically optional secondary tables or local textual strings), can be assigned in the non-reserved locations. Assignments should occur in sequence using the lowest possible open numbers to allow the receiving devices that have implemented an internal *table of tables* to point to each entry in its own local memory maps without wasted table entries.

TABLE 15—TABLE NUMBERING ASSIGNMENTS

Table Name	Type	Comments	Assigned Value
Alternative Names Table	Secondary	Optional use	0x09
Forward Referencing	Prime	Optional use	0x07
Grid table	Prime	Required, first entries defined local entries added and sent	0x03
Incident Modeling Table	Prime	Required if not all in line	0x05
Incident Numbering Table	Prime	Required if not all in line	0x06
Local Strings Table	Prime	Required if any local text is used	0x08
Model Tables	Prime	Required, first entries are defined local entries added and sent is used	0x04
National Numbers Table	Prime	Required, is created not sent	0x02
National Strings Table	Prime	Required, may not be sent	0x01
Private Text Strings	Prime	Optional use	0x0C
RDS Incident Encoding Table	Prime	Optional use	0x0D
Resource Availability Table	Prime	Optional use	0x0E
Table of Tables	Prime	Required	0x00
Temp Strings Table	Prime	Required, if not all in-line	0x0A
Textual Incident Encoding Table	Prime	Optional use	0x0B
Transit Adherence Encoding	Prime	Optional use	0x0F
Transit Schedule Encoding	Prime	Optional use	0x10
Weather Table	Prime	Optional use	0x11

7.1 Context Definition for: Binary Data

Use: This stream is used to transport binary data (such as a PKzip file) where the data is broken up into packets for transport, reassembled at the receiving end, and processed further. The internal context and use of the data is not defined here.

The table number entry defines where in the *table of tables* structure this data fits. A value of zero shall be used on data message which do not apply to any table entry. The “name” of the binary data can be considered a file name (complete with any extension as needed) for receiving systems which implement a file storage system. The name of the file shall be sent by setting both the packet count and the total packet values to zero to indicate that the payload contents will contain the “name” of the binary data to follow in a the local dialect as a data element of type *the_string*. It is the responsibility of the receiving device to arrange the name string into a suitable structure for local use.

Subtracting the bytes used by the context, provides the bytes remaining for payload. Consider a payload of ~200 bytes, this would allow a file of over 14 Megs to be transmitted. At 8 Kbs of throughput, this would require nearly 4 h to occur and is clearly a poor use of any wireless media. If large files are to be transmitted some host media have other modulation formats suitable for stationary uses which support higher data rates.

Like all streams transmitting an array of semi-static data over multiple frames, the context flags are presumed to remain constant over the entire set of frames. The revision value can be used to determine if the array contents have changed from the last time this stream was downloaded. Note that this implies that only one set of binary data (or one file) can be in the process of being transmitted at any given time. See Tables 16 and 17.

TABLE 16—BINARY DATA CONTEXT FLAG SETTINGS

Context Flags	Byte One	Byte One	Byte One
Bit	8 (MSB)	0	/ No encryption used
Bits	7 to 1	0	/ Reserved

TABLE 17—BINARY DATA CONTEXT TABLE ENTRIES

Context Table:	Size
Table_Num	1 byte
Revision	1 byte
Packet Count	2 bytes
Total Packets	2 bytes

7.2 Context Definition for: Expansion Tables (General Table Use)

7.2.1 USE—This stream provides the mechanism to send various expansion tables for use by payload data in the other streams to conserve bandwidth. It is primarily used to send compressed tables of textual elements used in string expansions. Note that some other tables (for example, the Grid Array Table) are specific instances of this table class and thus provide examples of this class in use. A number of specific use streams are derived from this basic stream type.

Most tables are comprised of variable length entries where rows and columns are delimited by either a 0x00 or a custom value. Sparse and full tables are supported. Multiple objects per index are supported. The various standard alphabets of symbols are supported. A fixed length entry variant is also supported for binary data content.

The tables supported are of three general types. First, and most common, with automated index values created by the encoder, is a list of objects (such as strings) where every subsequent index value has an entry. The index hashing table for this type of table can be automatically imputed by parsing the array and looking for the delimiter value which separates each variable length entry, hence the name Imputed Table. Receiver implementers can build such a table as desired. As the number of entries which fit in the previous packets can vary, a starting index context value is provided.

Another table type supported is the Sparse Table where index values are not continuous but are sparse in nature and hence a different storage implementation is usually employed. In this message format, sparse tables come in one or two ranges, 1~255 and 1~65 535. In this format, the index number entry precedes the table value entry.

A further variant of the tables allows for multiple variable length objects to be associated with each entry, each object delimited by a row delimiter character.

A further variant of the tables allows for fixed length binary entries (the meaning of the data is not specified, its use is presumed to be known by the specific stream and table number). In this format, the row delimiter character (in the context values section) is used as a byte count value. Any unused bits in the entry are rounded up to the next byte boundary and are transmitted with those bits set to zero.

In all cases, the set of table entries being transmitted must break on a row boundary to allow proper framing across the data frame boundaries. This effectively limits the length of any table to having entries around 20 bytes less than the host's native payload sizes.

The first index value (zero) of all tables is reserved and not used. Receiver implementers may wish to use this feature to store table specific header information in this area. In the decoding schemas, an index value of zero is taken to be a "toss back" token and does not produce a look-up into the table.

The use of imputed and sparse tables allows the easy construction of alternative names tables by transmitting the primary tables as an imputed table and the alternative names tables as alt tables in the sparse format. See the informative sections for an example of this.

The table number forms a unique ID for each table and is used to associate related sets of tables (secondary tables) to a prime table. A table may have one or more secondary tables. No sorting or look-up ordering is implied by this.

The tables can be used to send both binary and textual information. When sending textual information, the normal language encoding formats are supported. When sending variable length strings (either single objects or multiple) a choice of delimiters is supported for both rows and columns. If the standard delimiter is used (a 0x00 value) no alternative delimiter context table entry exists. If the table is a fixed length non-string type, then no delimiter is used and the next entry begins immediately following the specified number of bytes in the entry (specified by the row delimiter).

In summary, there are imputed index tables of single strings, multiple string objects, and fixed length fields (binary). There are also variations of these same tables for index ranges of where the index value is provided. By looking at the table number, the use of the table (for example, the local names table) can be determined. By looking at the revision value, the end user can determine if their own copy of the table is current.

Like all streams transmitting an array of semi-static data over multiple frames, the context flags are presumed to remain constant over the entire set of frames. The revision value can be used to determine if the array contents have changed from the last time this stream was downloaded. See Tables 18 and 19.

TABLE 18—EXPANSION TABLES CONTEXT FLAG SETTINGS

Context Flags	Byte One		
Bit	8 (MSB)	0	/ No encryption used
Bits	7 to 5		/ Table Type
		000	Index - Imputed, String I
		001	1~2 ¹⁶ Index Value. String
		010	Index - Imputed, Objects I,E
		011	1~255 Index Value, Objects D,E
		100	1~2 ¹⁶ Index Value, Objects D,E
		101	Index - Imputed, Fixed Length I,E
		110	1~255 Index Value, Fixed Length E
		111	1~2 ¹⁶ Index Value, Fixed Length E
Bits	4 to 3		/ Symbol Encoding
		00	Modified ASCII format
		01	Latin 1
		10	Unicode
		11	None, used for non-text tables
Bit	2		/ Table Type (Prime. Alt) H
		0	Prime
		1	Alternative
Bit	1		/ Column end delimiter value F
		0	Use 0x00 as std value
		1	Use custom table value

TABLE 19—EXPANSION TABLES CONTEXT TABLE ENTRIES

Context Table			
A	Revision	1 byte	
B	Packet Count	2 bytes	The payloads sequence number
C	Total Packets	2 bytes	
D	Object Count	1 byte	Count of objects in columns
E	Row Delimiter	1 byte	Token used to delimit rows or size (if used)
F	End Delimiter	1 byte	Token used to delimit end of columns
G	Table Number	1 byte	Unique Indent value
H	Prime Tbl Number	1 byte	Used if this table is an Alt to a prime
I	Imputed Indx	2 bytes	Used to start the imputed values for this frame
J	Total Entries Count	2 bytes	Used to convey last used table entry

7.3 Context Definition for: Forward Referencing Tables

- 7.3.1 USE—This stream is used to provide pointers and indexes into additional information related to an event (typically an incident traffic event but useable for other items such as current news stream) which can be obtained in other streams or from other media devices (access to which will vary based on user equipment). This method can also be used to provide limited advertising feedback by implementing a click-thought process using WWW URL pointers where the entry pointer tells the service provider from which original media the user was referred. The structures used is a 64K string table. The message itself is made up of table entries, each a three token stream. Token one is the index value. Token two (the first char of the stream) is the technology (and can be viewed as selecting the way to interpret the remainder). Token three is the string itself.

It is expected that the development and use of this stream will vary widely in different markets and host media depending of what public-private operating environment is developed. In any event, this stream should not be used to require the end user to seek critical data on these other media or streams. While this stream was intended to promote commercial interests and opportunities, it should not be used to the detriment of open and standardized ATIS content. See Tables 20 and 21.

TABLE 20—FORWARD REFERENCING CONTEXT FLAG SETTINGS

Context Flags:	Byte One			
Bit	8 (MSB)	0	/ No encryption used	
Bits	7 to 5		/ Table Type	
		001	1~2^16 Index Value. String	
Bits	4 to 3		/ Symbol Encoding	
		00	Modified ASCII format	
		01	Latin 1	
		10	Unicode	
		11	None, used for non-text tables	
Bit	2		/ Table Type (Prime. Alt)	H
		0	Prime	
		1	Alternative	
Bit	1		/ Column end delimiter value	F
		0	Use 0x00 as std value	
		1	Use custom table value	

TABLE 21—FORWARD REFERENCING CONTEXT TABLE ENTRIES

Context Table:			
A	Revision	1 byte	
B	Packet Count	2 bytes	The payloads sequence number
C	Total Packets	2 bytes	
F	End Delimiter	1 byte	Token used to delimit end of columns
G	Table Number	1 byte	Unique Indent value
H	Prime Tbl Number	1 byte	Used if this table is an Alt to a prime
I	Imputed Indx	2 bytes	Used to start the imputed values for this frame
J	Total Entries Count	2 bytes	Used to convey last used table entry

7.4 Context Definition for: Grid Array Tables

- 7.4.1 USE—This stream is used to provide the map “footer” anchor points from which the local Grid and tile system is expanded. It contains the data which is used by each local Grid map to in order to form a complete 32-bit lat/long value. It may be used to transmit zoomed Grids which exist on a much smaller or much larger size as required by the local application as well. It is a 255 fixed length table type. There will be no alternative to this table.

The flags for the zoom are taken over all entries in the table except the nationally defined entries. For example, if different zooms are used in the local entries this flag is set TRUE. The presence of local entries at the nominal zoom and national entries at another zoom does not cause this flag to be set true. The *non-standard* offset flag is set true if any tile is not at the standard offset rate or is not spaced at 50% of this rate (see 5.3 for details).

Like all streams transmitting an array of semi-static data over multiple frames, the context flags are presumed to remain constant over the entire set of frames. The revision value can be used to determine if the array contents have changed from the last time this stream was downloaded. See Tables 22 and 23.

TABLE 22—GRID ARRAY CONTEXT FLAG SETTINGS

Context Flags:			
Byte One—Derived from Expansion Tables Context			
Bit	8 (MSB)	0	/No encryption used
Bits	7 to 5	110	1~255 Index Value, Fixed length
Bits	4 to 3	11	None, used for non-text tbles/Symbol Encoding
Bit	2	0	/Table Type is Prime
Bit	1	0	/Column end delimiter value is Std
Byte Two—Grid Control Word			
Bit	8		/Variable Zoom Used
		0	NO
		1	YES
Bit	7		/Non-Std Offsets Used
		0	NO
		1	YES
Bit	6		/Reserved
		0	NO
		1	YES
Bit	5		/Vanity Maps in Use
		0	NO
		1	YES
Bits	4 to 1	0	/Reserved

TABLE 23—GRID ARRAY CONTEXT TABLE ENTRIES

Context Table:			
A	Revision	1 byte	
B	Packet Count	2 bytes	The payloads sequence number
C	Total Packets	2 bytes	
E	Row Delimiter	1 byte	Token used to delimit rows o size (if used)
G	Table Number	1 byte	Unique Indent value
J	Total Entries Count	2 bytes	Used to convey last used table entry

- 7.4.2 INITIAL TABLE ENTRIES—This stream has the first six entries predefined as a national zoomed Grid to support the description of events outside of the local areas but within the North American hemisphere. Refer to Section 12 - *Standardized National Tables Used in this document* for the definitions of these entries.

7.5 Context Definition for: Incident Flow Data

- 7.5.1 USE—This stream is used to describe most types of incidents (real, modeled, preplanned, etc.) and forms one of the most common streams of data used in ATIS. The items in the payload section contain numerous indexes into other related tables and data streams as needed to expand the message in a more verbose format. This stream uses the LRMS methods to refer to specific segments and then provides incident data on those segments (either by stating the current rates, by using a shared model, by a pointer to further textual based descriptions, or a combination of these).

The basic flow of messages in this stream are [<LRMS> <Names> <Model or Incident><other data>] where the format of elements themselves is self describing both by the flags below and in the stream itself (the MAYBE token).

The LMRS flags indicate the base type of element used. If for example the flags are set to *the_Grid* value ten, ALL values in the following messages will be in the Grid value. The most effective use of bandwidth is to set the token to mixed values thereby allowing the most efficient encoding to be determined in each case (point and delta first, then point and point, and finally a full lat/long when required).

The incident modeling flags allows for incidents to be transmitted in-line or to be referred to by table entry. Superposition can be turned on or off. Textual descriptions can also be by table or in-line. Textual description can be RDS based, table indexed, or free text. Various scalar values of flow rate can be transmitted in-line as well.

The naming structures also allow a wide set of variations. The optimal bandwidth is achieved with a datum of sufficient quality that no names need be transmitted to achieve coherence.³⁴ When names must be transmitted, each name is composed of three parts, the *pre*, *primary*, and *post* names. Pre- and post- name parts allow a special index into the national names table to exchange common entries such as "Street" effectively. For each part of a name string there are two related encoding options. It can be used or not, and then it can be limited to table entries (thereby prohibiting free text in this field), or not (thereby allowing free text in the field). These options are repeated for the cross street as well.

When the presence of a table entry is indicated by default in the header section, the token that normally selects which table is not preset in the payload area. For pre- and post-names, entries are limited to the first 255 index values of the national names table and is a 1 byte field. For index values used in the primary name a full 16-bit index is used (even if the index is < 255). Different tables are selected by the process of tables selection token (see Tables 7 and 8). Recursion in the Local Tables use of national entries is still allowed. If there is a need to transmit this type of data in the message, the flag must revert to the MAYBE bit and the stream will then have the normal table selection token present in this field.

Typically, the previous rules result in a name string which is composed of a pre-token, a primary string and a post token. Consider "North Drummond Street" which would be coded as a pre-token (0x8C = "North") a primary token with capitalization (0xEE064A = "drummond") a post token (0xD3 = "Street").

The context flags used in this flow can be set very ineffectively while still meeting the syntactically correct definition of the messages. To cite an extreme example, the flags could be set to provide uncompressed latitude and longitude values, complete with segment and cross street names in an uncompressed format (essentially achieving the geometry and cross streets profiles of the LRMS effort, with each item ~15 times bigger). Such a setting would produce such a large resulting message that effective information throughput would be reduced to a few segments per second. The most effective settings are typically to allow the encoder to select between NO (datum not sent) and MAYBE (mixed use of the datum) on a frame by frame basis. In any event, receivers should be prepared to process all valid settings which they may encounter. In this way, general text and tokens can be automatically optimized as processed. In total, this saves a considerable number of bits.

As a final example of efficient context flag usage, consider a typical street name which appears in the Local or National Tables. In an open stream of tokens, this name would be coded as three values "A-B,C-D,E". Where A-B is the combined exception token and the specific table (A indicating a table, B indicating which table, both coded in a single 8-bit value larger than 0x80) while C-D is a 2 byte index into this table, and finally E is the end of a string delimiter. By setting the proper bits, this expression may be reduced to just the index table selection (2 bytes) and the index value, saving 16-bits or 50%.

34. Bear in mind that the mapping of names and coordinates can be done by looking-up in the map table if the coordinates used are the same as sent here. Therefore, even if an end using device requires the name string to perform a full cross streets match, that data is extractable from the map and need not be transmitted with every update.

Note that some data elements will appear in pairs (example: cross street names will be either absent or appear for two cross streets). In these cases, the flag setting and any table entry applies to both data elements appearing in each item of the payload. See Tables 24 and 25.

TABLE 24—INCIDENT FLOW CONTEXT FLAG SETTINGS

Context Flags:			
Byte One - LRMS Formatting			
Bit	8 (MSB)	0	/ No encryption used
Bits	7 to 6		/ Lat - Long format used for pairs
			/ what type of LL will be used
		00	the _LL, 32-bit values
		01	the _Grid, 14-bit Grid pairs
		10	the _Grid plus the _delta
Bits	5 to 4	11	adjustable based on size
			/ Primary Segment name datum
		00	NOT used
		01	ALWAYS used
		10	MAYBE used
Bits	3 to 2		/ Cross Street Segments name datum
		00	NOT used
		01	ALWAYS used
		10	MAYBE used
Bits	1 (LSB)	0	/ Vertical Data Present
		0	NOT used
		1	MAYBE used
Byte Two - Naming Structures (set to all zeros if no Names)			
Bit	8 (MSB)		/ Prefix Name datum present
		0	NO, set to 0 if not used
Bit	7	1	YES
			/ Prefix Name value by table
Bit	6	0	MAYBE, uses modified ASCII format
		1	YES, is index value
Bit	5		/ Primary Name datum present
		0	NO, set to 0 if not used
Bit	4	1	YES
			/ Primary Name value by table
Bit	3	0	MAYBE, uses modified ASCII format
		1	YES, is index value
Bit	2		/ Post Name datum present
		0	NO, set to 0 if not used
Bit	1	1	YES
			/ Post Name value by table
Bits	2 to 1	0	MAYBE, uses modified ASCII format
		1	Yes, is index value
			/ Char Set used in these strings
		00	Modified ASCII
Bits		01	ISO Latin 1
		10	Unicode (2 bytes per char)
		11	Reserved
		Byte Three - Naming Structures (set to all zeros if no names)	
Bit	8 (MSB)		/ Prefix Cross-name datum present
		0	NO, set to 0 if not used
Bit	7	1	YES
			/ Prefix Cross-Name value by table
Bit		0	MAYBE, uses modified ASCII format
		1	YES, is index value

TABLE 24—INCIDENT FLOW CONTEXT FLAG SETTINGS (CONTINUED)

Context Flags:			
Bit	6	0	/ Primary Cross-name datum present NO, set to 0 if not used
		1	YES
Bit	5	0	/ Primary Cross-Name value by table MAYBE, uses modified ASCII format
		1	YES, is index value
Bit	4	0	/ Post Name datum present NO, set to 0 if not used
		1	YES
Bit	3	0	/ Post Cross-Name value by table MAYBE, uses modified ASCII format
		1	YES, is index value
Bits	2 to 1	00	/ Char Set used in these strings Modified ASCII
		01	ISO Latin 1
		10	Unicode (2 bytes per char)
		11	Reserved
Byte Four - Data and Model Formats			
Bit	8 (MSB)	0	/ Private Data at end of each message NO
		1	Yes (private data is in line, bit count by table value)
Bit	7	0	/ In line Data is in use NO (no byte five data will follow in stream)
		1	Yes (data will follow as per settings in byte 5 context)
Bits	6 to 5	00	/ Data mapping to segments Both IN and OUT links use one common set of data values, no mapping_seg in stream
		01	Mapping appears in the stream with mapping_seg value
		10	Reserved
		11	Reserved
Bits	4 to 3	00	/ In-Line Model Present NOT used (implies only in line data is present)
		01	Index value ALWAYS used (the table of models)
		10	In-Line model ALWAYS used
		11	Nothing and prior model (table or inline) now valid
Bits	2 to 1	00	/ Incident Events Description format None
		01	RDS Index Value to follow inline
		10	Tables Index value to follow inline
		11	String In Line (same encoding as the_string)
Byte Five - In Line Data Formats			
Bits	8 to 7	00	/ Link Speed Data In Line NOT used
		01	ALWAYS used
		10	MAYBE used
Bits	6 to 5	00	/ Link Occupancy Data In Line NOT used
		01	ALWAYS used
		10	MAYBE used
Bits	4 to 3	00	/ Link Travel Time, short value data In Line (0~180 s) NOT used
		01	ALWAYS used
		10	MAYBE used

TABLE 24—INCIDENT FLOW CONTEXT FLAG SETTINGS (CONTINUED)

Context Flags:			
Bits	2 to 1		/ Link Travel Time, long value Data In Line (0~4 h)
		00	NOT used
		01	ALWAYS used
		10	MAYBE used

TABLE 25—INCIDENT FLOW CONTEXT TABLE ENTRIES

Context Table:			
A	Grid Index	1 byte	/ the Grid used
	Issuing Time	2 bytes	/ the time for which offset models are issued
	Local Name Table	1 byte	/ the local segment table used
	Model Table	1 byte	/ the shared model table used by these MSG's
	Event Table	1 byte	/ the event stream for text and codes
	Private Event Table	1 byte	/ the event stream for additional text and codes

- 7.5.2 **RECOMMENDATIONS**—If in-line data is not being used (either discrete values or a shortened model) then no value data follows (only a model value will be found and possibly a description value and private data as well).

Data which follows may apply to the segment in both directions, either direction, or be two complete sets of data depending on the setting of bits 6 and 5 of byte four and stream value (mapping_seg) for each message.

The recommended setting to achieve the most effective encoding requires that efficient grid and offset lat/long values be used with a well processed set of name table entries and that historical modeling be used in conjunction with real-time incidents.

Encodings of this nature can provide incident flow update rates of over 100 incidents per second. In terms of the message payload, the resulting payload data entries would consist of sets of (see Table 26):

TABLE 26—EFFICIENT INCIDENT FORMATTING

Point	Delta	Model Number of Current Flow Rate
28 bits	16 bits	12 bits

All other data could be unambiguously derived from the tables as needed by the receiver device.

Alternatively, data can be provided in-line with each payload of messages carrying only those elements which are desired (for example sending only the flow information which would result in 28,16,8 = 48-bits per message). Further examples of such options are provided in the illustrative example sections.