TCIP V2.4

Draft Transit Communications Interface Profiles

Version 2.4

Acknowledgment

This document was drafted by APTA's TCIP Project Team. The authors would like to acknowledge the accomplishments and contributions of the TCIP I Technical Working Group and Subgroups which developed the NTCIP 1400-1408 Standards. This draft standard is built on the hard work, dedication, and products of that team. Without their efforts, this document would not have been possible.

This document was developed by APTA and two principal subcontractors ARINC Inc, and Critical Link, LLC under contract to the Federal Transit Administration. The APTA TCIP Executive Review Committee is chaired by John Bartosiewicz Executive Vice President/COO of McDonald Transit Associates, Inc., formerly the General Manager of Forth Worth Transit. Inquiries regarding this document should be directed to Louis Sanders of APTA (202) 496-4886.

Contents

List of Figures

List of Tables

1. Overview

The Transit Communications Interface Profile (TCIP) development effort began under the auspices of the Institute of Transportation Engineers (ITE) in cooperation with the U.S. Federal Transit Administration (FTA), the U.S. Federal Highway Administration (FHWA), and the ITS Joint Program Office. This development was performed in the context of the national Intelligent Transportation System (ITS) standards development effort and the National Transportation for ITS Protocols (NTCIP) effort of the American Association of State and Highway Transportation Officials (AASHTO), National Equipment and Manufacturing Association (NEMA) , and ITE. The purpose of this effort was to provide the transit standards to support the transit segment of the ITS National Architecture. This effort produced message formats and defined data elements for the exchange of information among transit computer systems in several business areas. More information on these standards can be found at: **www.its-standards.net**, **www.ntcip.org**, and **www.ite.org**. These standards are published by AASHTO, NEMA, and the ITE as:

NTCIP 1400	TCIP Framework Document
NTCIP 1401	TCIP Standard on Common Public Transportation (CPT) Objects
NTCIP 1402	TCIP Standard on Incident Management (IM) Objects
NTCIP 1403	TCIP Standard on Passenger Information (PI) Objects
NTCIP 1404	TCIP Standard on Scheduling/Runcutting (SCH) Objects
NTCIP 1405	TCIP Standard on Spatial Representation (SP) Objects
NTCIP 1406	TCIP Standard on On-Board (OB) Objects
NTCIP 1407	TCIP Standard on Control Center (CC) Objects
NTCIP 1408	TCIP Fare Collection (FC) Business Area Standard

The definition of TCIP "dialogs" which define the rules for sending and receiving these messages was not completed as a part of this initial effort. The dialogs are a key element in developing useful standard interfaces between transit business systems. These rules create relationships between messages and systems which are essential to successful real-time interfaces. For example, the rules in the dialogs specify how a server must respond to a request for information (such as a schedule), by returning a specified information message, or a specified error message to the requester.

After completion of the initial TCIP data element and message standards, APTA was invited by the FTA to take a more prominent role in TCIP development. This resulted in the development of this document, as well as a variety of tools to support TCIP implementers. These tools are available free of charge on the Internet at <<TBD URL>>>.

This document subsumes the work done to date in creating the 9 NTCIP Standards, and extends this work by adding additional data elements, data frames, messages and dialogs. This document also redefines conformance to allow for limited implementations scaled to agency and project needs, and to incorporate the use of batch transfers of transit information using TCIP messages in situations where agencies do not want or need the real-time information transfers specified in the dialogs.

Batch transfers allow standardized information to be transferred from one business system to another without a real-time or network connection. Batch transfers involve having one business system write a file containing one or more TCIP messages. The file is then moved to another system to be read.

Real-time information transfers are defined by dialogs. Dialogs define a sequence of TCIP messages that allow TCIP systems to interact without the need to manually move files form one system to another. Real-time interactions that begin using TCIP for batch transfers should account for the possibility of migrating to real-time connections as part of their planning process. Real-time information transfers occur over network connections, and may occur within very short time spans following an event that causes the information transfer. In other cases, real-time transfers may be deferred for operational reasons. For example, although a schedule for Tuesday of next week becomes available on Sunday, the CAD/AVL system may defer loading the schedule until Monday night.

2. Definitions

Table 2.0 provides definitions for terms used in this standard.

Table 2.0 Definitions

Term	Definition
AASHTO	American Association of State Highway and Transportation officials. A professional organization for transportation professionals. Develops and promotes transportation standards.
ADA accessible	Americans with Disabilities Act Accessible. The Americans with Disabilities Act requires public facilities in general, and transit facilities and services in particular to be made accessible to persons with disabilities. The definition of an accessible facility is an evolving one, as interpretations of the act change over time, and from locality to locality.
APTA	American Public Transportation Association. A non-profit association of transit agencies and suppliers of transit-related products and services.
ASN.1	Abstract Syntax Notation. A standard promulgated by the International Telecommunications Union as ITU-T X6.80 for "…defining the syntax of information data. It defines a number of simple data types, and specifies a notation for referencing those types, and for specifying values of those types.
ASN.1 Type	A data type defined by ASN.1 and used as a base type for defining TCIP data elements.
ATIS	Advanced Traveler Information System. A system for collecting and disseminating information to travelers, usually on a multimodal basis. Typical information provided includes transit, traffic, tourism and weather.
Attribute	A quality or characteristic inherent in, or ascribed to someone or something.
BIT STRING	A series of binary (1,0) values.
BOOLEAN	A variable whose values are limited to True and False.
Bound	A state of a vehicle or operator work assignment in which the work defined by the assignment has been associated with the specific vehicle or operator expected to perform the assignment.

Term	Definition
Canned Message	A text message which has been predefined, and stored in at least 2 locations, allowing the message to be specified for display to a human by transmitting its identifying number rather than by transmitting the entire message text. Some canned messages contain designated locations where canned text ("takes") can be inserted from a list (e.g. a bus stop name), into the message.
CC	Control Center. One of 8 business areas defined in TCIP I. Control Center activities include the dispatching, monitoring, controlling, and managing transit operations in real-time. Most TCIP CC activities involve interactions between the CAD/AVL and other fixed business systems and transit vehicles.
Centroid	1. The geometric center of an area 2. In Geographical Information Systems (GIS) terminology, the centroid is the point in a polygon linking information to that specific area.
Controller	In the control dialog pattern, there is a device which executes commands, and a device which issues commands to be executed. The controller is the command-issuer in those dialogs.
Controlled Device	In the control dialog pattern there is a device which executes commands and a device which issues commands to be executed. The controlled device is the command-executor in those dialogs.
Coordinated Universal Time	Time scale maintained by the Bureau International de l'Huere (International Time Bureau) that forms the basis of a coordinated dissemination of standard frequencies and time signals.
CPT	Common Public Transportation - One of 8 business areas defined by TCIPI. This business area involves the definition and distribution of information which is needed by several other business areas. Examples are transit facilities information, stop point lists, etc.
Data Element	An atomic piece of information which can be used in data frames and in messages. For example first name, speed, latitude, and footnote are data elements.
Data Frame	A grouping of data elements and/or data frames used to describe an object, that has complex attributes. For example a person has height, weight, gender, name , address, ect. as attributes.
Dialog Pattern	A description of a message exchange between two or more entities, including the rules associated with the exchange. Generally the same pattern can be used to convey more than one type of information- for example a subscription could convey schedules or alarm information.
Dispatch	The person responsible for sending out transit vehicles to operate according to schedule. Usually also the person who deals with exception conditions and incidents occurring during daily operations.
ENUMERATED	A variable whose value is restricted to a specified list of values. These values can be, but are not required to be numeric. In TCIP, enumerated types are typically assigned both token values (e.g. red, white, black) and corresponding numeric equivalents (1,2,3).
Event-Driven	A transaction or response in a transit business system or component that is based on the occurrence of an event. Events include human actions, vehicle movements, monitored parameter changes, received messages etc. The subscription dialog pattern provides for a subscription wherein updates to the subscriber are event-driven – meaning that updates are sent to the subscriber based on the occurrence of events.

Term	Definition
FHWA	Federal Highway Administration. A component of the United States Department of Transportation.
FTA	Federal Transit Administration. A component of the United States Department of Transportation.
GIF	Graphics Interchange Format. A file format for exchanging graphical images.
GIS	Geographical Information Systems. A computer software application that organizes and processes information based on geographical coordinates as well as other attributes.
Identifier	A unique number assigned to an item (bus, employee, stop point etc) to provide a short, and uniform way to reference that item, as distinct from all other items of the same type.
IM	Incident Management. One of the 8 business areas defined in TCIP I. Incident Management involves reporting, responding to, closing, and coordinating responses to events (incidents) that disrupt transit service.
INTEGER	A variable whose values are limited to zero, positive whole numbers, and negative whole numbers. TCIP integer types are further limited to a range of values that allow their storage requirements to be limited to 1,2 or 4 octets.
ITE	Institute of Transportation Engineers. A professional organization for transportation engineers. Develops and promulgates transportation standards.
ITS	Intelligent Transportation Systems. A worldwide initiative to use advanced technologies, and computer-based technologies specifically to enhance the safety and efficiency of transportation systems.
Logoff	An event wherein a user of a system or component notifies the system or component that an ongoing usage session is to be terminated.
Logon	An event wherein a user of a system or component notifies the system or component that a new usage session is to be initiated. Usually the logon includes the provision of the users identification number, and possibly security information.
LRMS	Location Referencing Message Specification.
Message	A grouping of data elements and/or data frames intended to be transmitted as a complete package of information in one direction.
Mobile Data Terminal	A device on a transit vehicle to allow the vehicle operator to exchange information with onboard components and systems. Interaction with external systems, e.g. a control center, may also be supported.
Mode	The type of transit service provided (e.g. bus, express bus, bus rapid transit, light rail, commuter rail, subway, etc.)
MPEG-4	A standard promulgated by the Motion Picture Entertainment Group for the distribution of full-motion video images.

Term	Definition
NEMA	National Electrical Manufacturers Association.
NTCIP	National Transportation Communications for ITS Protocol. A group of standards intended to promote interoperability among ITS components, and subsystems.
NULL	A value assigned to an item to indicate no value, or to signify that the item is not present.
Numeric String	A UTF8String consisting exclusively of the characters representing the digits zero through nine.
OB	On board. One of 8 business areas defined by TCIP I. On Board activities include interactions between components on a PTV.
OCTET STRING	A variable length sequence of octets. This is an ASN.1 base type used to convey unstructured blocks of data. Some of the data elements defined using this type conveys data which is further defined by another (non-its) standard. TCIP data elements defined using OCTET STRING as a base type always specifies a limitation on the upper length of the octet string.
Octet	A group of eight binary bits. Octets are a standard grouping for bits to be transmitted across a communications network.
PI	Passenger Information – One of 8 business areas defined by TCIP I. Passenger Information involves the creation, distribution and dissemination of information that will assist transit users (passengers) in efficiently using the transit system to plan and execute their travel.
PID	Passenger Information Display – An electronically changeable information sign that provides transit information to transit passengers. PIDs would typically be located at bus stops, transfer points and intermodal passenger facilities.
Pixels	The basic unit of the composition of an image on a television screen, computer monitor or other display.
Polygon	A feature used to represent geographic areas defined by the lines that form its boundary and a point (centroid) inside its boundary for identification.
Priority	Precedence established based on importance or urgency.
PTV	Public Transit Vehicle. Any vehicle used to provide public transit service.
Query	A request for information from one business system or component to another.
Queue	1. A line of waiting people, vehicles or other items. 2. A sequence of stored data, messages, programs, or events awaiting processing in a computer.
REAL	A variable whose values can include real number values.

Term	Definition
Repository	A business system in a transit agency, whose function is to accept and store the output of other business systems, and to make those results available to other business systems on demand. Some repositories may combine [fuse] data from different business systems and provide the results on request, or may process the data and provide the processed result to other business systems.
Roster	A grouping of operator work assignments into a weekly package.
Route	A path through a transit service area on which service is provided. Routes are generally maintained with a publicly known identifier which is used with all PTVs stop points and schedules related to service on the route. Route identifiers are often retained, even when modifications are made to the service maintained along the route.
SCH	Scheduling. One of the 8 business areas defined by TCIP I. Scheduling involves the creation and distribution of transit schedules, vehicle and operator assignments, rosters, span of service and other schedule-related information.
SEQUENCE	A keyword used in ASN.1 to indicate that an object (data frame, or message in TCIP) consists of a sequence of other objects (data frames and/or data elements in TCIP).
Shape points	Geographical locations that define points along a line, other than its termini, provided for the purpose of defining the shape of the line as other than a straight line.
SCP	Signal Control Prioritization. A TCIP Business Area added.
SP	Spatial Positioning. One of 8 business areas defined by TCIP. Spatial Positioning defines how locations, geographical boundaries, areas, addresses etc are defined. SP also defines some units of measure which are used by other business areas. Effective in TCIP version 2.4 SP data elements and data frames will be based on the (LRMS) standard to the extent feasible.
State Plane	A coordinate system used in mapping the United States. It divides all fifty states, Puerto Rico, and the Virgin Islands into 120 numbered zones, and each zone number defines the projection parameters for the region.
Stoppoint	A location where PTVs normally stop to allow passengers to board and/or alight.
Subscriber	A business system or component that requests and accepts data from another.
Subscription	A relationship between an information consumer (subscriber) and an information provider (server) in which a contract (subscription) is established governing the information transfer. In TCIP there are three types of subscriptions – query, event, and periodic. These provide information on a one-time-basis, as-changed-basis, and recurring-interval-basis respectively.
Termini	The "terminating" or end nodes of a line, link, or route.
Text Take	A segment of text intended to be selected from a list of similar text takes to be inserted into a larger text message. For example a list of text takes representing bus stop names, would be used as the source for the bus stop name to be inserted into a bus stop arrival announcement ("*Now approaching Broad Street Station*")

Term	Definition
Transfer	1. A location where two or more transit routes provide service to the same or very closely located stop points, allowing passengers to conveniently switch between service provided on the various routes. 2. An event in a passenger's trip itinerary wherein the passenger must alight from one PTV and board another to switch from service on one route to service on another route.
Trigger	1. The initiation of a series of actions (as of a dialog). 2. The event that initiates a series of actions.
Trip	A specified series of movements of a transit vehicle between an origin and a destination end point. Trips in revenue service normally involve passing through defined timepoints and providing service at stop points.
Unbound	A state of a vehicle or operator work assignment in which the work defined by the assignment has not been associated with the specific vehicle or operator expected to perform the assignment.
UTF8String	A variable capable of conveying a series of characters, numbers, and special characters. This data type is capable of conveying special character sets, although this is not recommended for TCIP. Refer to ISO 8824-1.
Vehicle Logic Unit	A computer on a transit vehicle that coordinates a variety of operational functions usually including Automatic Vehicle Location and messaging.
XML	Extended Markup Language. W3C (Which defines Internet standards) defines XML as the "universal format for structured documents and data on the web".
XML Schema	Defines the structure of XML documents (or messages in TCIP) W3C states "XML schemas express shared vocabularies and allow machines to carry out rules made by people. They provide a means for defining the content and semantics of XML documents".

3. Conformance

Two types of conformant interfaces are defined for TCIP: TCIP Batch Interfaces, and TCIP Real-Time Interfaces. Batch interfaces are intended to allow agencies to benefit from the use of TCIP message structures to convey information in the form of files between transit operating systems, usually under manual control. Real-time interfaces are intended to allow agencies to connect transit operating systems and facilitate automated network-based transfer of information between connected systems in a structured manner. Real-time interfaces do not always imply immediate transfers of data. For example schedule changes may be created during the day and downloaded to the CAD/AVL system through the network on a nightly basis. Batch processes involve vendor defined mechanisms for generating and exchanging files and thus do not ensure interoperability.

A TCIP Batch interface is conformant if it uses a TCIP message format inside a computer file as a medium to exchange information. A conformant batch interface must comply with all specifications, rules and constraints specified in:

- the message definition for the message being used for the batch transfer,
- data frame definitions for all data frames required in the message,
- data element definitions for all data elements in the message and included data frames

A TCIP Real-time interface is conformant if it uses one or more TCIP dialogs as a medium to exchange information. A conformant real-time interface must comply with all specifications, rules and constraints specified in:

- the dialog definition,
- the dialog pattern definition associated with the dialog,
- message definitions for all messages required in the dialog,
- data frame definitions for all data frames required in the messages,
- data element definitions for all data elements in all messages and data frames required in the dialog

4. Understanding TCIP

TCIP standardizes information exchanges among transit business systems and components, either on a batch (file transfer), or real-time (message exchange) basis. By standardizing, and modularizing these interfaces the intent is to minimize the cost of tailoring interfaces. These costs occur as a result of vendor changes to interfaces for an individual agency, or another supplier. By standardizing these interfaces, different agencies and vendors will be able to reuse the same interfaces. By modularizing these interfaces, each agency will be able to select the specific interactions across the interfaces (dialogs) necessary to meet their specific business needs.

TCIP defines the messages, data frames, and data elements to be used for information transfer on either a batch or real-time interface. For real-time interfaces, TCIP also defines the sequence of interactions between the interfaced systems in the form of dialogs. TCIP implementations are modular so that each agency can choose the specific standard messages to be exchanged on a batch basis, or the specific dialogs to be implemented on a real-time basis to meet that agency's needs. These decisions should be documented in the agency's architecture along with network and protocol definitions and other information allowing them to serve as a baseline and a starting point for RFP development. Figure 4.0 depicts the building block approach to TCIP. Each building block 'sits' on the other building blocks on which it depends.

The lowest level objects in TCIP are dialog patterns and data elements. Dialog patterns define a type of exchange in generic form, such as the sequence of actions in a query or a command-response. Data elements define the lowest level data objects in TCIP and how they are represented. For example a data element CPT-EmployeeFirstName is defined as having type NAME, which is in turn defined to be a string of up to 30 characters.

TCIP data frames build on TCIP data elements by grouping data elements together to describe something meaningful in the real world. For example an employee description might include other items besides first name such as middle and last name, phone number, employee number and address.

TCIP messages build on TCIP data elements and data frames, incorporating them into a meaningful one-way information transfer package. For example a message defining an incident might incorporate information concerning the employee reporting the incident, vehicles involved, people injured, incorporate as many data elements and data frames as are required to create a complete one-way information transfer.

TCIP dialogs build on the dialog patterns which define the structure of the interaction (such as a query) and the messages which define the content of the interaction. (the information to be exchanged). TCIP batch builds on the messages, but not the dialog patterns, because batch interactions are vendor-defined.

The agency architecture builds on the agency's legacy systems- on TCIP-Batch (if used) and on the TCIP dialogs. The agency architecture also needs to consider how the agency fits into the regional architecture, and how the agency and other regional stakeholders plan to evolve their business systems over time.

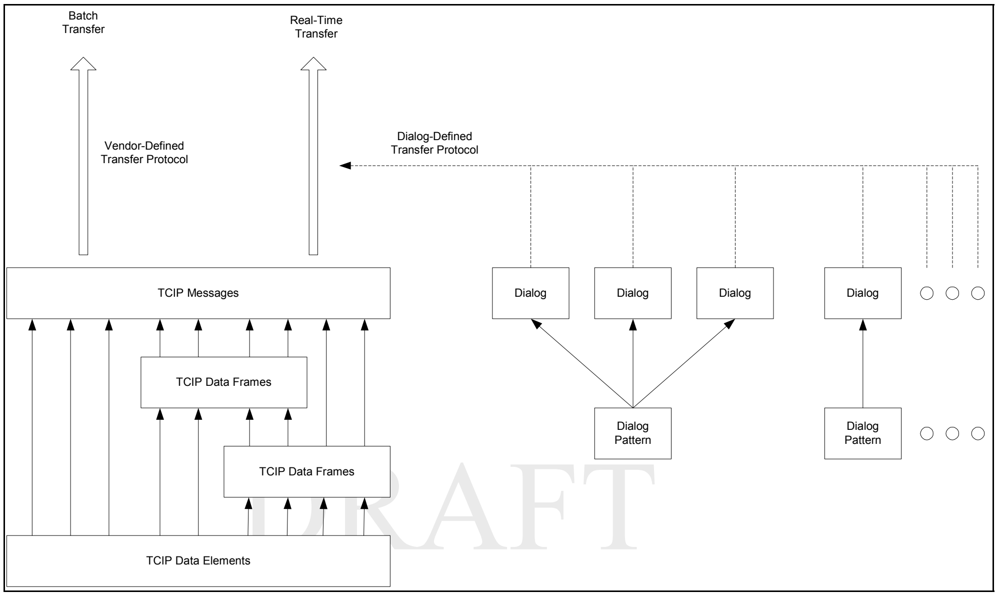


Figure 4.0 depicts the relationships among TCIP components.

5. Concept of Operations

5.1 General Concepts

TCIP provides a standardized framework for information exchange between:

- Transit agency business systems and components within a transit agency,
- Transit agency business systems in different transit agencies,
- Transit agency business systems and outside agency business systems

TCIP is designed to minimize the impact on transit agency operating paradigms, and on existing products and systems. TCIP allows a transit agency to implement TCIP on an incremental basis by implementing only the dialogs required to meet business requirements at any point in time rather than requiring wholesale conversion of all business systems to TCIP. Thus agencies can maintain legacy non-TCIP systems and interfaces in place alongside TCIP systems and interfaces, while achieving compliance with TCIP and the ITS National Architecture.

TCIP is designed to minimize the impact on supplier systems and products as well. TCIP does not specify interactions within the components produced by suppliers, or between computer applications and users. For example, if a user requests a trip itinerary from a passenger information system, TCIP does not specify the screens, user interactions, etc. TCIP does provide the data structures and dialogs to facilitate the passenger information system obtaining schedule information from the scheduling system. TCIP also provides data structures and dialogs to allow one passenger information system to provide itinerary information to another (e.g. to another agency). TCIP uses extended markup language (XML) to provide a widely-known and supportable data exchange format between business systems.

In practice TCIP provides the tools for a transit agency to select the standard information flows required to meet its business needs, and to implement those flows cost efficiently. Information flows are standardized by defining the message formats for the exchange, as well as the dialogs that define how sequences of messages are used to implement an information flow.

The procuring transit agency, after selecting the appropriate information flows for its needs, specifies what new and existing systems are to play what roles in what dialogs. For example a passenger information system may be designated a client, and a schedule repository a server for an information flow to convey availability of scheduled service. The procuring agency also determines what systems are authorized to access what information, and specifies what systems are responsible for controlling access. TCIP does not specify the security procedures used to validate an information request, but does provide error messages to allow an unauthorized request to be rejected.

An agency architecture defines the systems and interfaces within a transit agency as well as the interfaces between agency and other agencies and/or private entities. The architecture is the repository for capturing and documenting the agency's legacy interfaces as well as the TCIP interfaces. The agency architecture may specify a series of phases. Each phase would represent upgrades or replacements of existing systems and interfaces, or new systems and interfaces being added. In each phase the specification of the interfaces would call out the TCIP dialogs required to be implemented by that interface and/or the legacy data flows to be maintained across the interface.

5.2 Common Public Transportation

N/A to TWG 10

5.3 Incident Management

N/A to TWG 10

5.4 Passenger Information

N/A to TWG 10

5.5 Scheduling/Runcutting

N/A to TWG 10

5.6 Spatial Representation

N/A to TWG 10

5.7 On board Vehicles

N/A to TWG 10

5.8 Control Center

N/A to TWG 10

5.9 Fare Collection

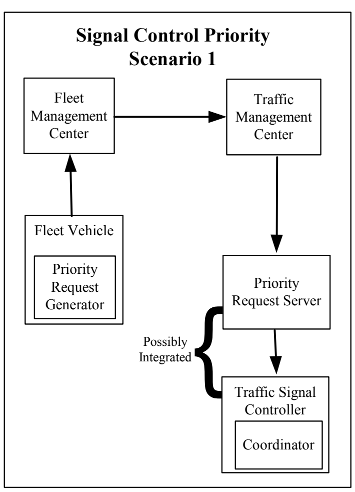
N/A to TWG 10

5.10 Signal Control and Prioritization

This business area is concerned with obtaining preferential treatment from arterial traffic signal devices for transit vehicles, and for managing information related to interactions between transit vehicles, the transit control center and traffic signal system and its associated traffic management center. In general transit vehicles receive precedence from the traffic signal system, rather than signal preemption, which is generally reserved for emergency vehicles. Precedence provides for reduced wait times for transit vehicles while mitigating the disruption to the coordinated operation of traffic signals caused by preemption.

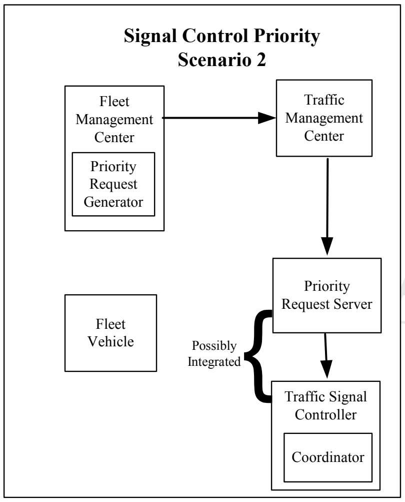
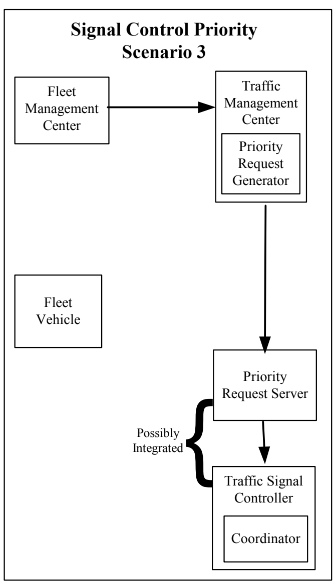
Interactions with the traffic signals to request, update, status and cancel priority requests for transit vehicles are governed by NTCIP 1211 "Object Definitions for Signal Control and Prioritization". NTCIP uses the Simple Network Management Protocol (SNMP) as defined by the Internet Engineering Task Force (IETF) of the Internet Activities Board (IAB). For clarity some information from these documents has been incorporated into TCIP, however, these sources take precedence in the case of any omissions, or differences, between them and this document.

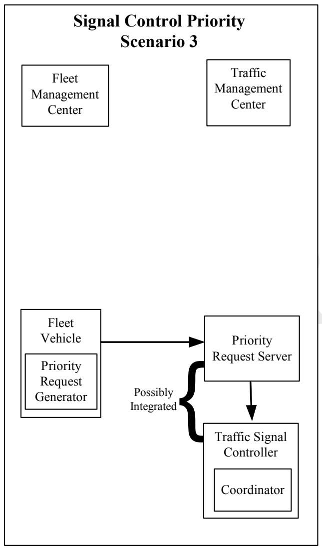
NTCIP 1211 defines four scenarios for transit signal priority operations based on the location of the transit Priority Request Generator (PRG) and the path used to get to the traffic signal system's Priority Request Server (PRS). The four scenarios defined by NTCIP 1211 are depicted in table 5.10. Scenarios 1,2, and 4 are based on the transit-based PRG generating 6 types of messages to the Priority Request Server: Priority Request, Priority Update, Priority Status Control, Priority Status Buffer, Priority Cancel and Priority Clear. Scenario 3 requires the transit control center to send operating information to a Traffic Management Center (TMC) based PRG, which in turn generates the 6 types of messages to the PRS.

NTCIP 1211 Signal Precedence Scenarios		
Table 5.10		
Scenario	Figure	Description
1		The transit fleet vehicle carries a Priority Request Generator (PRG). Priority requests are send through the transit fleet management center (control center) to the Traffic Management Center and onward to the Priority Request server in the field for processing. Response messages follow the reverse path.

NTCIP 1211 Signal Precedence Scenarios

Table 5.10

Scenario	Figure	Description
2		The Transit Fleet management Center (control center) generates Priority requests which are sent through the Traffic Management Center to the Priority Request Server. Response messages follow the reverse path. The Transit Vehicle is not directly involved in the exchange.
3		The transit fleet management center (control center) sends information about the transit fleet operations to the Traffic Management Center which generates priority requests to the Priority Request Server. Neither the transit vehicles nor the Transit control center are directly involved in the processing of priority requests or responses.

NTCIP 1211 Signal Precedence Scenarios		
Table 5.10		
Scenario	Figure	Description
4		The transit fleet vehicle carries a Priority Request Generator (PRG). Priority requests are sent directly to the Priority Request server in the field. Response messages follow the reverse path. Neither the transit control center, nor the Traffic Management Center are directly involved in the processing of priority requests or responses.

5.10.1 Transit Management/Traffic Management SCP Planning

A Transit SCP Program requires extensive coordination with the Traffic Management Community. The specific corridors and signals to be included, the strategies to be employed at those signals need to be coordinated. This includes specific strategy number assignments, and agreements about where they are to be used, as well as coordination of approaches to near versus far side stops, queue jumping lanes, transit only signal phases etc. In addition, numerous technical issues need to be decided early in the process including the specific NTCIP 1211 scenario(s) to be employed, the communications network(s) to be employed and address assignments.

5.10.1.1 Coordination of NTCIP 1211 Scenarios

Selection of NTCIP 1211 Scenarios for use in SCP planning is driven primarily by the planned or existing infrastructure on the vehicle at the transit control center, at the intersection, at the Traffic Management Center, and for communications. While selection of a single scenario for use throughout an agency simplifies the design and management of SCP, local constraints may dictate the use of different scenarios at different intersections and or in different jurisdictions.

Scenarios 1 and 4 rely on the availability of computer resources on the transit vehicle as well as wireless communications from the bus to the control center or the intersection respectively. The existence of wireless networks is not sufficient to determine that one of these scenarios is appropriate, however, as the availability of capacity on the networks and network response times must be evaluated to verify that they are adequate to support SCP.

Scenarios 2 and 3 do not require wireless communications form the bus to deliver priority requests, but they do require the control center to be cognizant of current bus locations and progress against schedule. The accuracy and timeliness of this information is critical to the effective implementation of SCP.

Scenarios 1,2, and 3 all depend on a sufficiently robust, and high-capacity communications connection between the transit-control center and the traffic management center. Realistic evaluations of the performance and capacity of the infrastructure and early coordination of infrastructure requirements and scenarios to be used is critical to the success of an SCP program.

5.10.1.2 Coordination of SCP Strategies

While a variety of SCP strategies have been defined, local coordination will determine what strategies are to actually be implemented under what conditions in a given corridor or intersection. These decisions are constrained by the street topology and lane structures, capabilities of the intersection equipment and controllers, and locally established policies.

Strategies available for consideration include:

- Transit-only lanes to facilitate queue jumping
- Transit-only phases
- Truncation of opposing green phases to create an early green
- Extension of a green phase to wait for a transit-vehicle arrival
- Pedestrian phase eliminating for a cycle
- Phase rotation

Once the strategies to be employed are agreed between the transit and traffic management group, the operational conditions under which these strategies are to be used must be determined. For example, a locality may determine that once a transit priority is granted at an intersection, no more transit requests can be accommodated for a fixed number of cycles. Another example is that priority requests might only be allowed for vehicles in revenue service, or that are running late. Some strategies or combinations of strategies may not be available to a transit vehicle under all circumstances – for example elimination of pedestrian phases might not be allowed for lightly loaded buses.

5.10.1.3 Example of SCP Strategy Coordination

This section is an illustrative example of how strategies might be coordinated and applied, it is not necessarily appropriate for any particular locality.

An example locality might determine that the only transit priority strategies to employ are;
T-Early Truncation of Opposing Green with Early Extension of Requested Green
R-Phase Rotation to provide the requested green earlier
P- Pedestrian Elimination form the current cycle
E-Extend Green phase to accommodate an arriving vehicle.

Based on further coordination, the agencies agree that the operating conditions under which the strategies may be employed are as shown in Table 5.10.1.3-1.

Table 5.10.1.3-1 Allowed Strategies

Passenger Loading Schedule Status	Light	Medium	Heavy
Early	None	None	None
On time	R	R	R
Slightly Late	R+E	R+E	R+E
Late	R+E	R+P+E	R+P+E
Very Late	R+P+E	T+R+P+E	T+R+P+E
Too Late	None	R	R

The rationale was that:
1. Early buses do not need priority treatment
2. Phase Rotation is the least disruptive, and therefore the first strategy offered

3. Extended Greens are the next to least disruptive and the next strategy offered
4. Pedestrian Phase Elimination and Early Truncation are to be avoided, except there the bus is already late or very late
5. If the bus reaches a threshold of lateness, and cannot be expected to recover, it will be offered less help

Based on this decision the transit agency decides to quantitatively define the fuzzy logic terms in Table 5.10.1.3-1 as shown in Table 5.10.1.3-2.

Table 5.10.1.3-2 Allowed Strategies Quantified

Passengers Loading/Schedule Status		Light 0-10 passengers	Medium 11-25 passengers	Heavy >25 passengers
Early	<0 seconds	None	None	None
On time	0-60 seconds	R	R	R
Slightly Late	61-120 seconds	R+E	R+E	R+E
Late	121-600 seconds	R+E	R+P+E	R+P+E
Very Late	601-1200 seconds	R+P+E	T+R+P+E	T+R+P+E
Too Late	>1200 seconds	None	R	R

The Traffic Management Agency decides to adopt a systemic approach to assigning strategy numbers within the corridor receiving the SCP upgrade. The range of strategy numbers 50-250 are reserved for transit. The agency decides to allocate strategies using the same algorithm for all intersections in the corridor.

An additive algorithm is chosen. The algorithm has 3 components:

- D – The direction from which the vehicle approached the intersection
- S - The strategies to be requested for a given priority request
- A – The angular path through the intersection (Straight, Left Turn, Right Turn)

Values are assigned for D, as follows

- Approach intersection from the North (D=50)
- Approach intersection from the South (D=100)
- Approach intersection from the East (D=150)
- Approach intersection from the West (D=200)

Values assigned for S based on the list of strategy types requested. Thus if two types of strategy are requested S is the sum of the values assigned for those strategy types. The strategy type values were assigned as:

T-Early Truncation (1)
R-Phase Rotation (2)
P-Pedestrian Phase Elimination (4)
E-Extend green Phase (8)

Thus if all four strategy types are requested, S=15. If Extended Green and Phase Rotation only are requested, S=10.

Finally A is assigned based on the vehicle's intended path through the intersection as follows:

Straight thru intersection – A=0
Left Turn at intersection – A=16
Right Turn at intersection – A=32

Based on this approach, a transit vehicle's expected intersection approach direction, requested priority strategies, and path through the intersection are uniquely determined. This approach is for illustrative purposes only and is not applicable to all localities or intersections.

5.10.2 TCIP Signal Control & Prioritization (SCP) Dialogs

TCIP defines dialogs and messages to facilitate operations in each of the four NTCIP 1211 scenarios. The dialog "Subscribe SCP History Data" provides a mechanism for a transit control center or data repository to obtain a history of SCP events (from the traffic management center's viewpoint) from a suitably equipped Traffic Management Center, this dialog is applicable to all 4 NTCIP 1211 scenarios.

5.10.2.1 SCP Scenario 1 Dialogs

In this scenario, a Priority Request Generator onboard the vehicle initiates priority requests which are sent through the transit control center and the Traffic Management Center to reach the Priority Request Server.

The dialog "Upload SCP Data" provides a mechanism to upload the data required by a bus to interact with SCP-equipped intersections projected to be on its routes throughout the data. The data includes a default distance from an intersection to initiate a priority request, a list of intersections with stop bar locations, optional overriding distances at which to initiate prioritization, addressing information and strategies for each intersection. It also includes criteria that establish schedule adherence parameters passenger loading parameters and operational status that must be met to select a strategy.

The dialog "Download SCP Performance Data" provides a mechanism to download the record of all SCP events (from the vehicle point of view) from the vehicle to the control center or data repository.

The dialog "SCP Priority Request Scenario 1" provides a mechanism to obtain SCP precedence in accordance with NTCIP 1211 Scenario 1.

5.10.2.2 SCP Scenario 2 Dialogs

In this scenario, the vehicle is not directly involved in the generation of SCP Priority requests. The dialog "SCP Priority Request Scenario 2" provides the mechanism for a Priority Request Generator in the transit control center to obtain SCP precedence on behalf of a transit vehicle. In practice a transit control center would need to simultaneously execute multiple instances of this dialog to service the transit vehicle fleet.

5.10.2.3 SCP Scenario 3 Dialogs

In this scenario, neither the vehicle nor the transit control center contains the Priority Request Generator. Instead the Traffic Management Center performs this function on behalf of the transit fleet. The TMC requires information about the operation of the transit fleet in order to perform this function. The dialog "Subscribe Fleet SCP Information" provides a mechanism to transfer this information. The TMC sends a subscription request that contains the stop bar locations, of the SCP equipped intersections along with intersection address information and direction of travel for the approach to each stop bar. The Transit Control Center provides an event-driven subscription service that sends information about transit vehicles approaching stop bars that are candidates for priority request generation. The control center may optionally determine that some transit vehicles are not to be given priority treatment and withhold information about those vehicles from the TMC.

This dialog does not provide real-time feedback to the control center about priority requests generated or granted, however historical information about these requests can be obtained from a TMC using the "Subscribe SCP History Data" dialog, as with other SCP scenarios.

5.10.2.4 SCP Scenario 4 Dialogs

In this scenario, a Priority Request Generator onboard the vehicle initiates priority requests, which are sent via wireless communications network directly to the Priority Request Server.

As with Scenario 1, the dialog "Upload SCP Data" provides a mechanism to upload the data required by a bus to interact with SCP-equipped intersections projected to be on its routes throughout the data. The data includes a default distance from an intersection to initiate a priority request, a list of intersections with stop bar locations, optional overriding distances at which to initiate prioritization, addressing information and strategies for each intersection. It also includes criteria that establish schedule adherence parameters passenger loading parameters and operational status that must be met to select a strategy.

The dialog "Download SCP Performance Data" provides a mechanism to download the record of all SCP events (from the vehicle point of view) from the vehicle to the control center or data repository.

The dialog "SCP Priority Request Scenario 4" provides a mechanism to obtain SCP precedence in accordance with NTCIP 1211 Scenario 4.

5.10.3 SCP Operational Examples

This section provides examples to illustrate the operation of the SCP business area. Figure 5.10.3 depicts the route map for the examples in the subsequent sections.

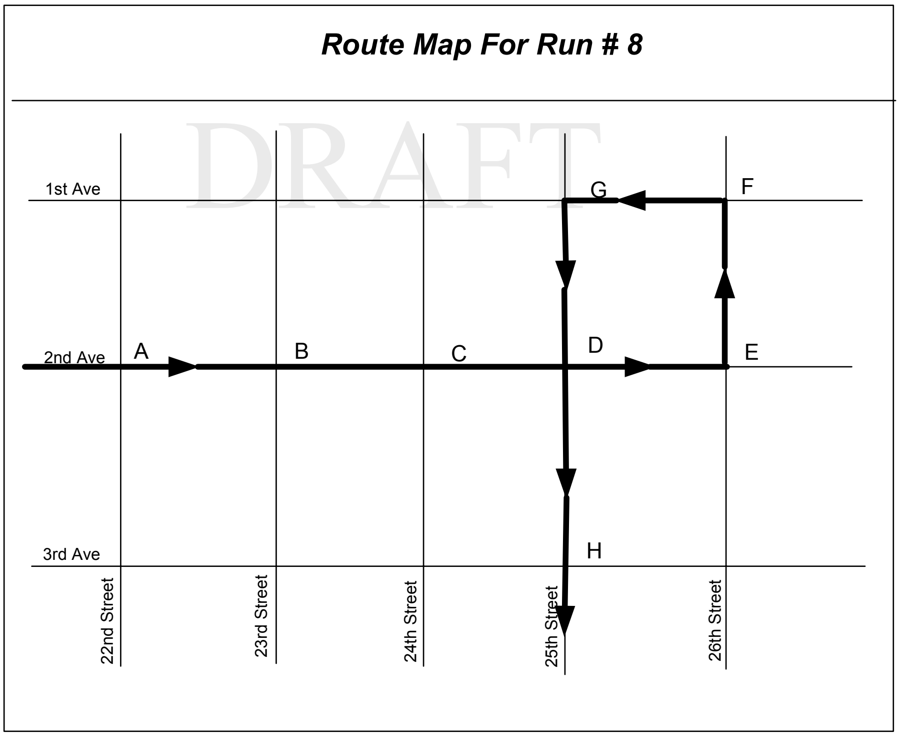


Figure 5.10.3

5.10.3.1 Vehicle Based PRG Scenario

This section illustrates the use of the information uploaded to a transit vehicle via the "Upload SCP Data" dialog by an onboard Priority Request Generator. It is applicable to NTCIP 1211 Scenarios 1 and 4. In this illustration a local decision has been made not to explicitly "clear" each intersection. Thus the illustration provides no examples of a Priority Clear, similarly, a local decision not to send status requests results in the PRG not sending Priority Status Control or Priority Buffer messages to the PRS.

An onboard PRG is assumed to have access to real-time data contained in the bus's vehicle logic unit. This information includes schedule information, real-time location information, service status (in-service, enroute,to start of route, etc.), and passenger loading (if the bus is equipped with passenger counters). Intersection information loaded to the bus is keyed to the travel patterns provided in the schedule upload.

Prior to leaving the garage area the bus is uploaded its SCP data along with other operational information required for its daily block of work. Table 5.10.3.1-1 depicts a subset of this information. The criteria and strategies used in this table are consistent with the example in section 5.10.1.2. The map in Figure 5.10.3 depicts the routing of the bus for Run #8 as used in this example. Table 5.10.3.1 shows all of the Priority Request Generator events at each intersection in this example. The remaining paragraphs in this section provide a narrative of the events at SCP-equipped intersections on the run.

Table 5.10.3.1.1
"Upload SCP Data" Provided Information

Default Decision Distance				200M						
Activation Date				8/18/04						
Deactivation Date				8/22/04						
Intersection Address		A	B	C	D	E	F	G	D	H
Run #		8	8	8	8	8	8	8	8	8
Pass #		1	1	1	1	1	1	1	2	1
Stop Bar Location		LatLon	LatLon	LatLon	LatLon	LatLon	LatLon	LatLon	LatLon	LatLon
Decision Distance		-	-	-	-500m	-	-	-	-	-
Criteria # 1	Earliest	0	0	0	0	0	1	1	1	1
	Latest	60	60	60	60	60	60	60	60	60
	MinLoad	0	0	0	0	0	0	0	0	0
	MaxLoad	100	100	100	100	100	100	100	100	100
	Strategy	202	202	202	202	218	118	168	52	52
Criteria # 2	Earliest	61	61	61	61	61	61	61	61	61
	Latest	120	120	120	120	120	120	120	120	120
	MinLoad	0	0	0	0	0	0	0	0	0
	MaxLoad	100	100	100	100	100	100	100	100	100
	Strategy	210	210	210	210	226	126	176	60	60
Criteria # 3	Earliest	121	121	121	121	121	121	121	121	121
	Latest	600	600	600	600	600	600	600	600	600
	MinLoad	0	0	0	0	0	0	0	0	0
	MaxLoad	10	10	10	10	10	10	10	10	10
	Strategy	210	210	210	210	226	126	176	60	60
Criteria # 4	Earliest	121	121	121	121	121	121	121	121	121
	Latest	600	600	600	600	600	600	600	600	600
	MinLoad	11	11	11	11	11	11	11	11	11
	MaxLoad	100	100	100	100	100	100	100	100	100
	Strategy	214	214	214	214	230	130	180	64	64
Criteria # 5	Earliest	121	121	121	121	121	121	121	121	121
	Latest	1200	1200	1200	1200	1200	1200	1200	1200	1200
	MinLoad	11	11	11	11	11	11	11	11	11
	MaxLoad	100	100	100	100	100	100	100	100	100
	Strategy	214	214	214	214	230	130	180	64	64
Criteria # 6	Earliest	601	601	601	601	601	601	601	601	601
	Latest	1200	1200	1200	1200	1200	1200	1200	1200	1200
	MinLoad	0	0	0	0	0	0	0	0	0
	MaxLoad	100	100	100	100	100	100	100	100	100
	Strategy	215	215	215	215	231	131	181	65	65
Criteria # 7	Earliest	1201	1201	1201	1201	1201	1201	1201	1201	1201
	Latest	100000	100000	100000	100000	100000	100000	100000	100000	100000
	MinLoad	11	11	11	11	11	11	11	11	11
	MaxLoad	100	100	100	100	100	100	100	100	100
	Strategy	202	202	202	202	218	118	168	52	52

As the bus approaches intersection A, which is the first SCP–capable intersection in this run, the PRG determines its schedule adherence status and the number of passengers onboard. Assuming that there are 8 passengers onboard, and the bus is on time, the PRG will select criteria #1 as matching its current operational conditions. Since there is no unique decision distance specified for intersection A, the PRG will initiate the priority request to the PRS upon reaching the default decision distance from the intersection A stop bar (200m).

As the bus moves between intersection A and B, two additional passenger board, and the bus "picks up" thirty seconds so that it is now running 30-seconds early. Upon reaching the default decision distance 200 meters from the stop bar, the Priority Request Generator determines that no criteria in the table match the current operating conditions, and therefore generates no priority request.

As the bus approaches intersection C, there are still 10 passengers onboard, however the bus has lost time in traffic and is now running 30-seconds late. Upon reaching the default decision distance of 200 meters from the stop bar, the Priority Request Generator determines that criteria #1 matches its current operating status and generates a priority request to the PRS with strategy number 202 requested. The bus stops 100 meters from the stop bar for one minute. During this period the PRG determines that the bus will not arrive at the stop bar as predicted and sends a Priority Cancel to the PRS.

As the bus approaches intersection D for the first time, it is 3 minutes late, and has 9 passengers onboard. The Priority Request Generator uses the intersection-D, Pass-1 column in the table for this instance. Since this column has an overriding decision distance of 500m, the Priority Request Generator initiates a Priority Request to the PRS with a requested strategy of 210 at a distance of 500 meters from the stop bar. As the bus continues to approach the stop bar, the bus has "made-up" 15 seconds, and sends a Priority update message to the PRS.

As the bus approaches intersection E, it is 2 1/2 minutes late, and has 18 passengers aboard. At 200 meters from the stop bar the PRG on the bus determines that criteria #4 applies, and sends a priority request for strategy #230.

SCP Priority Requests Table
Table 5.10.3.1-2

Run	Intersection	Pass	Schedule Adherence	Passengers Aboard	Intersection Direction	Criteria Number	Strategy
8	A	1	0	8	west/straight	1	202
8	B	1	-30	10	west/straight	none	0
8	C	1	+30	10	west/straight	1	202
8	C	1			west/straight		0/cancel
8	D	1	+180	9	west/straight	3	210
8	D	1	+165	9	west/straight	3	2/0 Update
8	E	1	+150	18	west/left	4	230
8	F	1	+90	25	south/left	2	121
8	G	1	+60	20	east/left	1	168
8	D	2	+30	30	north/straight	1	61
8	H	1	+30	5	north/straight	1	61

As the bus approaches intersection F, it has "made-up" a minute and now has 25 passengers aboard and selects criteria #2 and a strategy number of 121. At the default distance of 200 meters from the stop bar, the PRG sends the priority request to the PRS.

As the bus approaches intersections G, D (second pass), and H it is less than a minute late and chooses criteria #1 in each case. The PRG requests strategies 168, 61, and 61 respectively at a distance of 200 meters from each stop bar.

5.10.3.2 Control Center-Based Priority Request Generator

This example describes operations in NTCIP 1211 Scenario 2. In this scenario, the Control Center contains the Priority Request Generator, and generates priority requests on behalf of the transit fleet, through the Traffic Management Center. The criteria table and route information used in section 5.10.3.1 are reused in this section as the data set for the Control Center-based PRG.

In this scenario, the Control Center may not have the granularity of bus location information to generate priority requests at a clearly designated approach distance to the stop bar, as with the previous example. The bus's passenger loading may not be known to the Control Center, and changes to the bus's schedule adherence status may not be known to the Control Center in real-time. Table 5.10.3.2 summarizes the PRG activities in this example.

SCP Priority Requests Table Table 5.10.3.2							
Run	**Intersection**	**Pass**	**Schedule/ Adherence**	**Passengers Aboard**	**Intersection Direction**	**Criteria Number**	**Strategy**
8	A	1	0	8	west/straight	1	202
8	B	1	-15	10	west/straight	none	0
8	C	1	0	10	west/straight	1	202
8	D	1	+59	10	west/straight	1	210
8	D	1	+130	4	west/straight		cancel
8	E	1	+138	10	west/left	3	226
8	E	1	+150	18	west/left	4	230/update
8	F	1	+90	25	south/left	2	121
8	G	1	+60	20	east/left	1	168
8	D	2	+30	30	north/straight	1	61
8	H	1	+30	5	north/straight	1	61

As the bus approaches intersection A, it sends a location report to the Control Center. Location reporting to the Control Center is specified by the "Subscribe PTV-AVL" dialog. The Control Center's PRG forecasts the vehicle's arrival time at the default decision distance of 200m from the stop bar, and determines that another location report is not expected prior to the bus's arrival at the decision point. As in section 5.10.3.1 the PRG determines that criteria #1, strategy 202 applies and sends a priority request to the PRS via the Traffic Management Center.

As the bus approaches intersection B, the transit Control Center determines that a particular location report will be the last before reaching the decision point. As of this location report the bus is 15 seconds early. The Control Center PRG generates no priority request for intersection B. the bus continues to gain time and arrives at intersection B 30 seconds early.

As the bus approaches intersection C, the Control Center receives a location report indicating that the bus is on time, and determines that this is the last location report before reaching the decision point 200 meters in advance of the stop bar. The Control Center PRG matches the bus's state with criteria #1 and requests strategy #202 in a priority request to the PRS through the TMC. The bus continues to lose time as it approaches the decision point and arrives 30 seconds late at intersection C.

With the bus approaching intersection D, the Control Center receives a location report showing the bus to be 59 seconds late and to have 10 passengers aboard. The PRG determines that the bus will arrive at the decision point 500 meters in advance of at the stop bar before the next location report and sends a priority request using criteria #1 and a strategy number of 202, to the PRS via the TMC. Before reaching the intersection the bus sends a new location report indicating that it is now 130 seconds late, stopped and has 4 passengers aboard. The Control Center determines that is cannot accurately forecast the bus's arrival time at intersection D and the Control Center's PRG sends a cancel priority request to the PRS via the TMC. The bus leaves intersection D 180 seconds behind schedule.

Approaching intersection E the bus sends a location report indicating that the bus has 10 passengers aboard and is 138 seconds behind schedule. The Control Center determines that the next location report is due just past the decision point. The Control Center's PRG selects criteria #3 and strategy #226, and sends a priority request to the PRS via the TMC. Later the Control Center receives another location report indicating that the bus is 150 seconds behind schedule, 10 meters before the decision point and has 18 passengers aboard. The Control Center's PRG sends a priority update to the PRS via the TMC. The priority update provides a revised arrival projection at the stop bar and a revised strategy based on criteria #4, strategy #230.

As the bus approaches intersection F, it has "made-up" a minute and now has 25 passengers aboard. The Control Center's PRG, upon receiving the last location report prior to the decision point, sends a priority request via the TMC to the PRS. Based on a match with criteria #2, the request is for strategy #121.

As the bus approaches intersections G, D (second pass), and H it is less than a minute late. As the last location report prior to each decision point is received, the control center's PRG matches criteria #1 and sends priority requests with strategies 168, 61, and 61 respectively to the PRS via the TMC.

5.10.3.3 Traffic Management Center (TMC)-Based Priority Request Generator

This example describes operations in NTCIP 1211 Scenario 3. In this scenario, the Transit Control Center sends operational information to the Traffic Management Center (TMC) where the Priority Request Generator (PRG) generates priority requests to the appropriate Priority Request Server (PRS). The example route in section 5.10.3 is reused in this section.

The TMC establishes a subscription to the Control Center's fleet information using he "Subscribe Fleet SCP Information" dialog, this subscription is established prior to any signal priority events. The subscription allows the Control Center to send vehicle SCP information to the TMC for vehicles determined by the Control Center to be priority eligible. This information includes stop bar location, vehicle id, ETA at stop bar, vehicle type and vehicle loading. Similarly the Control Center subscribes to each vehicle's current location information using the "Subscribe PTV-AVL" dialog. These subscriptions are assumed to be in effect throughout this example.

As the bus approaches intersection A on route 8, the Control Center receives a location report and predicts a stop bar arrival time for intersection A. The Control Center determines that the bus is on schedule and sends vehicle SCP information to the TMC/PRG.

As the bus approaches intersection B, the Control Center receives a location report and determines that the bus is early, and sends no information to the TMC.

As the bus approaches intersection C, the Control Center receives a location report, determines that the bus is 30 seconds late, predicts a stop bar arrival time, and sends vehicle SCP information to the TMC/PRG. Subsequently, the bus stops at a location 100 meters prior to the stop bar for one minute, and sends another location report. The Control Center sends a new set of vehicle SCP information to the TMC/PRG. The PRG in the TMC decides whether to request a cancel or to update the previous priority request.

As the bus approaches intersection D for the first time, the Control Center receives a location report indicating that the bus in 59 seconds late, and has 10 passengers aboard. The Control Center sends vehicle SCP information to the TMC/PRG. Before reaching the stop bar, the bus sends another location report, and it is now determined to be 130 seconds late with 4 passengers onboard. The Control Center sends new vehicle SCP information to the TMC/PRG. The PRG determines whether to update or cancel the request.

As the bus approaches intersection E, the Control Center receives a location report indicating that the bus has 10 passengers aboard and is 138 seconds behind schedule. The Control Center determines that another location report will arrive with the bus still in advance of the intersection and sends no information to the TMC/PRG. Subsequently another location report arrives indicating that the bus is 150 seconds behind schedule, with 18 passengers aboard and 210 meters prior to the stop bar. The Control Center sends vehicle SCP information to the TMC/PRG.

CC to TMC/PRG Vehicle SCP Information Table 5.10.3.3					
Run	Intersection	Pass	Schedule Adherence	Passengers Aboard	Intersection Direction
8	A	1	0	8	west/straight
8	B	1	-15	10	west/straight
8	C	1	+30	10	west/straight
8	C	1	+90	10	west/straight
8	D	1	+59	10	west/straight
8	D	1	+130	4	west/straight
8	E	1	+138	10	west/left
8	E	1	+150	18	west/left
8	F	1	+90	25	south/left
8	G	1	+60	20	east/left
8	D	2	+30	30	north/straight
8	H	1	+30	5	north/straight

As the bus approaches intersections F, G, D (second pass), and H, the Control Center sends vehicle SCP information to the TMC/PRG for processing in each case. Table 5.10.3.3 summarizes Control Center to TMC/PRG interactions for this example.

5.10.1.4 Coordination of Vehicle Classes

NTCIP 1211 specifies that priority requests include a vehicle class and level value, each in the range 1..10. The effect of these values is to allow vehicles of a higher class to override the priority requests of vehicle of a lower class. Vehicles of the same class, but a higher level are given precedence within the class, but cannot override a request within the class.

The definition of the vehicle class and level assignments is a local decision. TCIP provides mechanisms to allow class and level assignments for transit vehicles based on vehicle type (e.g. fixed route bus, charter bus, supervisory vehicle), vehicle status (in-service, enroute to go into service, ferrying between garages etc), passenger loading, and schedule adherence. TCIP also provides a range of locally defined criteria values to allow other locally determined policies to be created to control class and level assignments.

5.10.1.5 Coordination or Priority Clear

NTCIP 1211 provides a definition of a message exchange to allow a priority request generator for a vehicle to notify the intersection's priority request server when the vehicle has cleared an intersection. The server will however time out and clear the priority entry from its tables if no clear indication is received.

TCIP provides for configuration of the priority request generator to explicitly clear intersections, or not, on an intersection by intersection basis. This allows a local decision to be made, with intersection level granularity as is to whether an explicit priority clear should be generated. The trade off here is that configuring the priority generator not to explicitly clear conserved communications network bandwidth, but delays the clearing of priority entries from the server's table.

6. TCIP Dialogs

TCIP dialogs define the message sequences exchanged between transit business systems and/or components to achieve a specific information transfer or operational objective. The dialogs define what messages or conditions initiate, maintain, and terminate the dialog. The dialogs defined in this document are intended to be simple and modular. Simple dialogs perform a single discrete purpose, and specify the minimum possible about the internal workings of the entities involved in the dialog. Modular dialogs are able to be used in combination to achieve complex interactions or singly to perform their assigned function only. Because they are modular, each agency can specify the dialogs they want implemented, and the role that each agency business system is to perform in each dialog.

These dialogs specifically exclude the internal actions within the transit agency business systems including:

- how data is stored, translated and manipulated,
- how data is formatted and presented to human users,
- what triggers (either manual or automated) cause the systems to initiate dialog(s),
- how systems and/or components trigger or initiate dialogs,
- the details of the interactions of components and systems with human users.

6.1 Patterns

Patterns define a dialog in a generic format that can be reused for multiple purposes. For example a periodic subscription dialog pattern allows a client entity to obtain information from a server entity on a periodic basis without querying for it each time. This pattern can then be used for multiple disparate purposes such as determining the current locations of specific vehicles periodically, or determining the value of a vehicle health parameter periodically. The use of a small number of patterns to implement a much larger body of dialogs simplifies the implementation process when compared with the approach of creating and standardizing unique control flows for every operational situation.

6.2 Dialog Instantiations

Dialogs are the highest level of abstraction in TCIP. A dialog specifies an operational purpose, a dialog pattern to be used, the messages to be used with the specified pattern, and any special conditions or constraints associated with the implementation of this specific dialog. A dialog may also specify relationships with other dialogs. For instance a query dialog that provides a specified version of the stop patterns associated with a route, could indicate that the user can determine the correct version of the patterns to request by using another dialog prior to sending the pattern query.

6.3 TCIP Batch

Some agencies do not have the direct connections between their business systems, but still have a need to exchange information between systems. A mechanism to transfer data between systems without a real-time messaging is through the use of files. One computer application saves the data in a file, the file is then transferred (possibly manually) to another system where it is loaded and read by another computer application. Such interactions require an agreed-upon data format for the file(s) to be exchanged. TCIP messages provide such a format. The XML format of TCIP messages allows the messages to be saved in a text file. Text files are readable by virtually any type of computer system. TCIP dialogs are not required for batch data exchanges. Vendor-specific procedures govern the process for causing the files to be generated by one application and to be read by another application.

Because batch transactions involve vendor defined interactions to generate, move, and read in files, they do not ensure interoperable implementations. TCIP-based batch processing is, however, a useful intermediate step to allow an agency to convert its systems to standard data formats in preparation for the implementation of dialogs. TCIP dialogs do provide the basis for interoperable systems.

7. TCIP Dialog Patterns

7.1 Subscription Pattern

A subscription dialog is a three (3) message dialog pattern. The pattern defines the conversation between a client and a server. The server is the owner/producer/provider of information required by the client who subscribes to the information. The three messages are a subscription request, a subscription response, and an error notice. The subscription request message and the subscription response message are unique to the individual dialog, however both contain a CPT-SubscriptionHeader data frame. The error message is common to all subscription dialogs and is defined as CptSubErrorNotice. Any failure to communicate, or to receive an expected response must be recovered by a vendor specified process within the component.

The subscription request message name has the form AaaXxxSub where:

- Aaa indicates the standard (e.g. Cpt, Sch) where the message is defined.
- Xxx indicates the name of the subscription (e.g. RouteList).
- Sub indicates that the message is a subscription request message.

The subscription response message name has the form AaaXxx where:

- Aaa indicates the standard (e.g. Cpt, Sch) where the message is defined.
- Xxx indicates the name of the subscription (e.g. RouteList).

The CPT-SubscriptionHeader contains a data element defined as CPT-SubscriptionType which allows a subscription dialog to assume 3 forms:

- The first form is a basic query. The client requests the information and the server provides it on a one-time basis.
- The second form is the periodic subscription. The client requests the information and the server provides it initially and on a recurring basis at intervals specified in the subscription request.
- The third form is the event-driven subscription. The client requests the information and the server provides it initially and provides new versions of the information based on events that change the information.

The CPT-SubscriptionType also provides the capability for the client to cancel a periodic or event-driven subscription. In the event that the server receives a subscription request message (AaaXxxSub) with a subscription header data frame indicating that the subscription is to be cancelled, the server shall cease sending the matching AaaXxx messages for the indicated subscription(s). Note that the cancellation message may request the cancellation of a single dialog only with a matching subscription number, or all dialogs for that subscriber of type AaaXxx.

The CPT-SubscriptionHeader provides identifiers for the subscriber (client) and the host (server) in the fields subscriberIdentifier and hostIdentifier. These fields are of the type CPT-ApplicationID, which is an agency-assigned unique identifier for computer applications within the agency's architecture. Agency's will need to coordinate with other agencies with whom they share data to ensure uniqueness of these identifiers (e.g. within a metropolitan area).

In the event that a periodic, or event driven subscription request arrives at a server and the expiration time for the request has already passed, the server shall immediately downgrade the request to a basic query, provide a single response to the client (assuming the request is valid) and end the dialog. In the event that the expired request is also invalid for other reasons, the server shall respond with a CptSubError notice.

Figures 7.1.A through 7.1.C depict the variations of the subscription pattern.

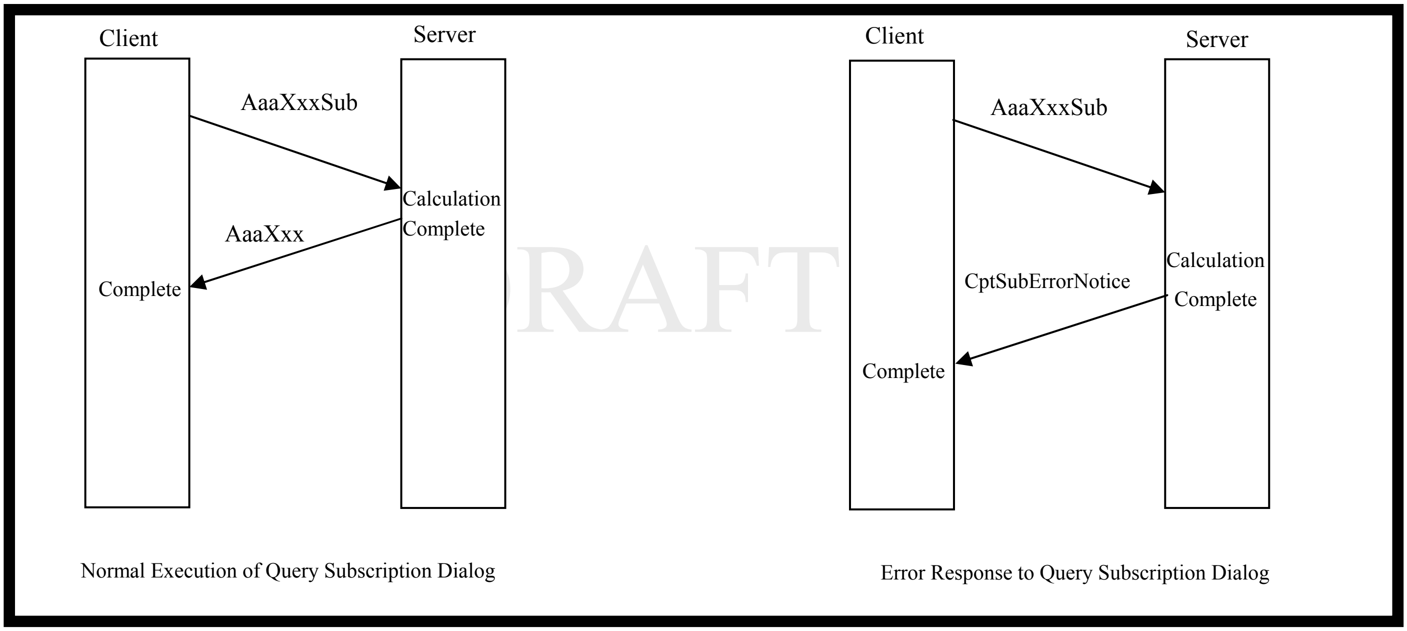


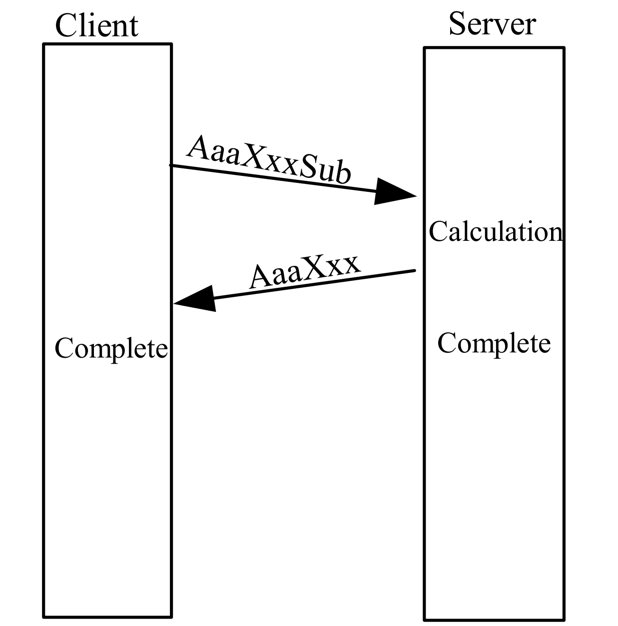
Figure 7.1.A Query Version of the Subscription Dialog Pattern

Although the client (subscriber) requests the form of the query in CPT-SubscriptionHeader, the server (provider) may change (downgrade) the request in the response. The allowed changes by the provider are:

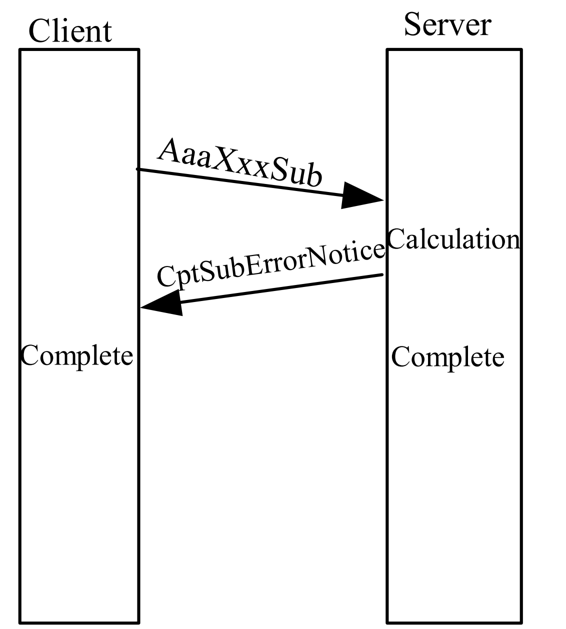
- Periodic request changed to Query response.
- Event-Driven Request changed to Query Response.
- Periodic Request changed to a longer reporting interval. For example if a subscriber requested information updates every second to the bus stop inventory the provider might change it to daily or weekly.
- Subscription expiration date/time changed to an earlier date/time. For example if the subscriber requested a 100 year subscription, the provider might limit the subscription to six months.

In addition to the request information provided in CPT-SubscriptionHeader, the subscription request generally contains additional specifications on the data to be provided that is dialog-specific. There may be dialog-specific limitations on the data in the response. For example if the planned schedule is requested for the next year, but has only been defined for the next six months, the provider might have a dialog specific option to reduce the scope to the data that actually exists.

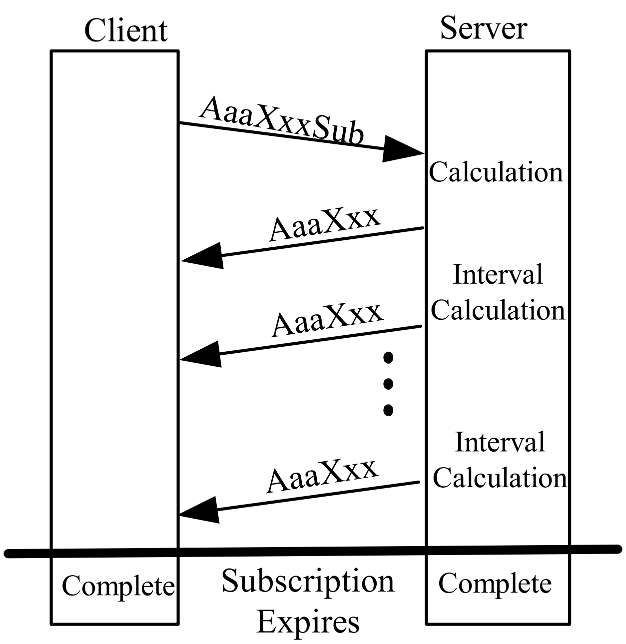
A procuring agency may limit the allowable downgrades for any specific subscription dialog. For example, if an agency is procuring a passenger information system that subscribes to bus locations on a periodic basis, the agency may require the AVL system provider to service the periodic subscription, and not downgrade it to a query.



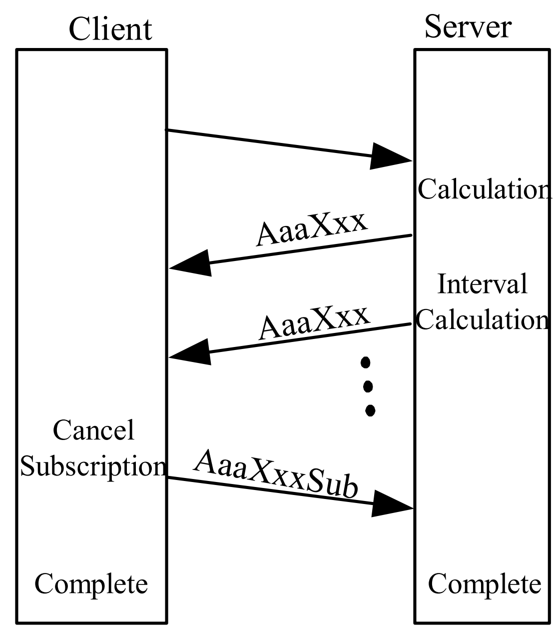
Query Downgrade of Periodic Subscription Dialog



Error Response to Periodic Subscription Dialog

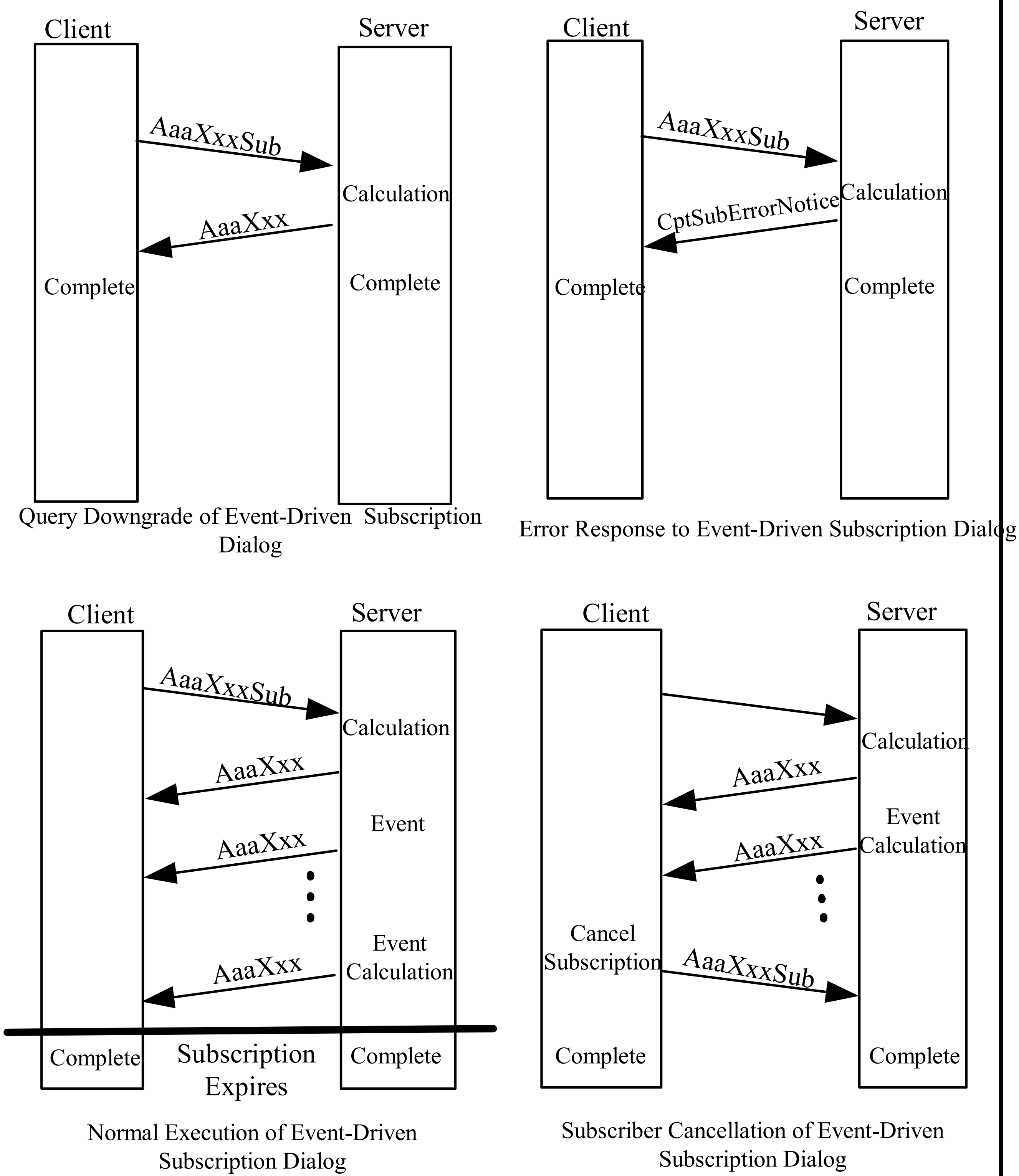


Normal Execution of Periodic Subscription Dialog



Subscriber Cancellation of Periodic Subscription Dialog

Periodic Version of the Subscription Dialog
Pattern Figure 9.1.B



Event Driven Version of the Subscription Dialog Pattern Figure 9.1.C

7.2 Command-Response Pattern

A Command Response dialog is a two (2) message pattern. The pattern defines the conversation between a controlling entity (controller) and a controlled entity (device). The device performs actions or provides services in response to commands from the controller. The two messages are a command message and a response message.

The controller initiates all command response dialogs by sending a command message. The usual case is that the device attempts to perform the command and replies with a response message indicating whether or not the command was executed successfully. A command identifier number of type CPT-CommandID assigned by the controller and conveyed to the device in the command message is returned to the controller in the response message allowing the controller to match the response with the previously issued command.

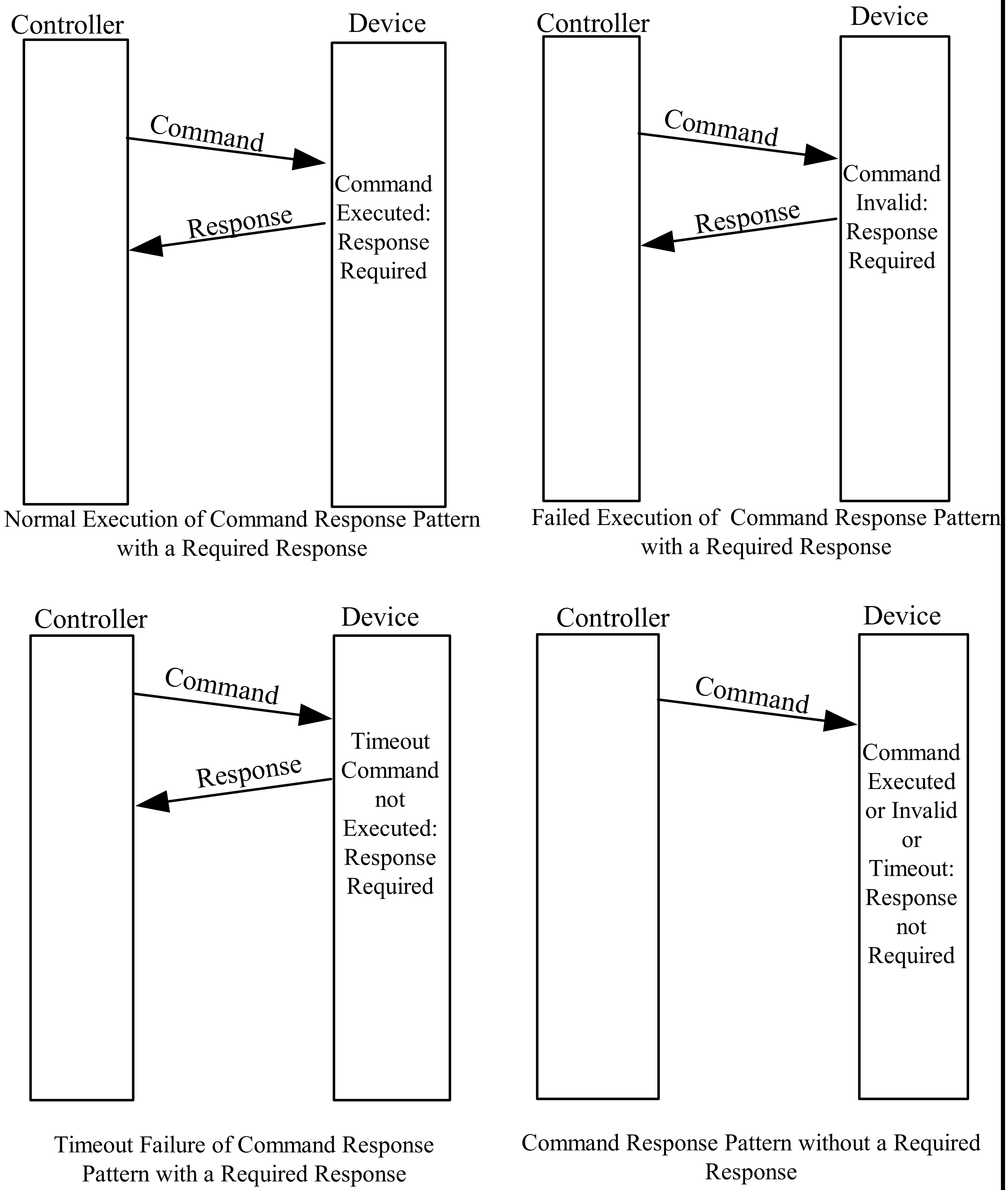
Some dialogs specify the use of the Command Response dialog pattern with one of two variations

- In the first variation the dialog specifies that the device only sends a response if a designated field in the command message asks for a specific response. Fields of the type are generally of type CPT-Boolean and have names like verifyCommand, or responseRequired.

- In the second variation, the dialog specifies that the device never sends a response, in this case a corresponding response message will not be specified for that dialog.

Some dialogs may specify that the command must be executed within a specified time frame or abandoned. The time frames may be specified in the dialog definition or in the command message. Such dialogs will specify a response message containing a data element of type CPT-ErrorCode which can be used to indicate that the command was not executed due to a timeout.

Figure 7.2 depicts the executionof the Comand Response dialog pattern.



Command Response Dialog Pattern
Figure 9.2

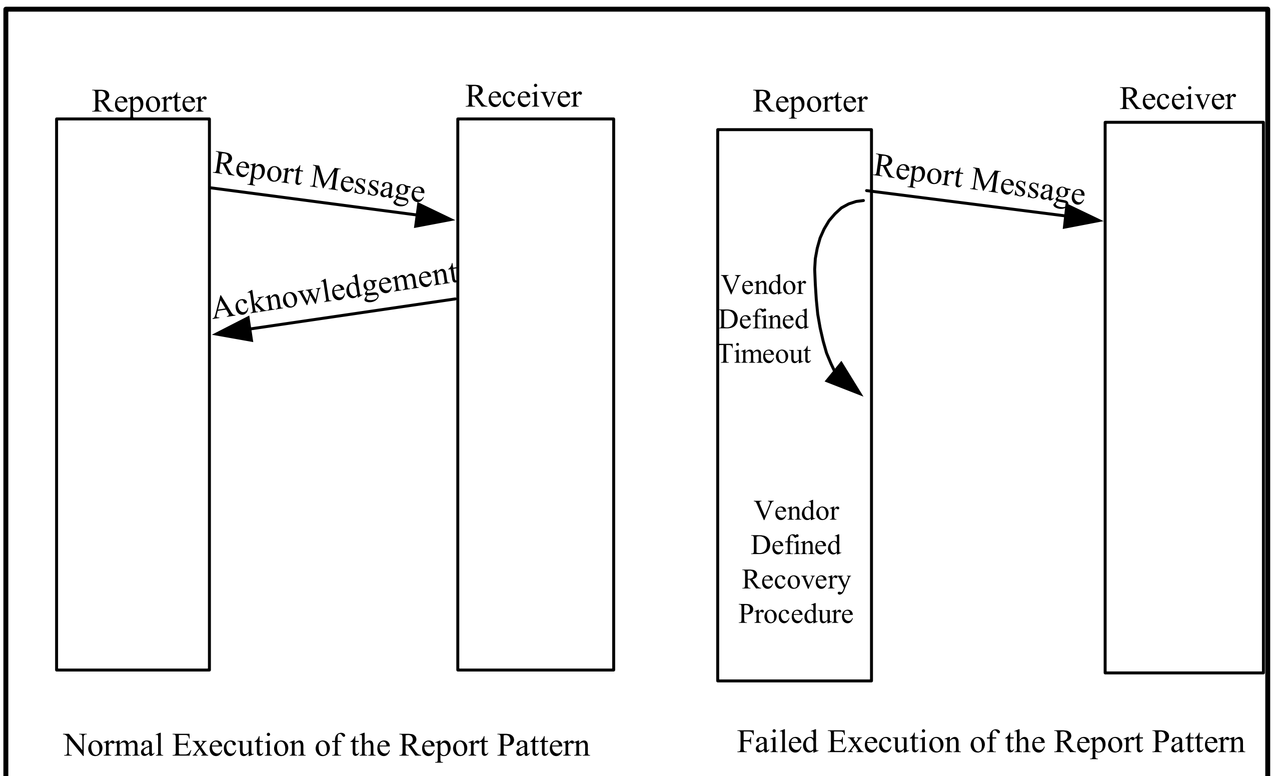
7.3 Report Pattern

A Report Dialog is a two (2) message pattern. The pattern defines the conversation between a reporter entity, (such as a field, or mobile based employee computer), and a receiving entity such as a supervisor hand held device, dispatch system, or maintenance management system.

The reporter entity generates the report based on actual conditions, or human input, not based on a query or subscription request from the receiving entity. Because reports are generated on an ad-hoc basis and from a variety of sources, the reports are given unique identifiers by combining their source identifier with the date-time of the report.

The report pattern is not intended to broadcast notifications to a wide distribution list, and the receiver is required to acknowledge receipt of the report. Vendor/Agency defined recovery procedures are executed by the reporter if the acknowledgement is not received by the reporter. Such procedures may include retrying the transmission, generating an alarm, notifying the operator (if applicable), or aborting the dialog.

Figure 7.3 depicts the execution of the Repor dialog pattern.



Report Dialog Pattern
Figure 9.3

7.4 Silent Alarm Pattern

The Silent Alarm pattern is a unique pattern specifically intended for use with transit vehicles with a silent alarm feature. The dialog is initiated by the vehicle computer when the alarm is triggered, and the vehicle computer sends a ImSilentAlarm message to the dispatch computer system. The vehicle computer may perform other agency/vendor defined actions as a result of the trigger.

After an agency/vendor defined notification and acknowledgment between the dispatch computer and the dispatcher, the dispatch computer provides an acknowledgment (ImSilentAlarmAck) to the vehicle. The dispatcher and/or the dispatch computer may initiate other vendor/agency defined actions as a result of the notifications. For example, this may include changing the location reporting update rate for the vehicle. Upon receipt of the acknowledgement, the vehicle computer uses a vendor/agency defined covert signal to notify the driver that the acknowledgement has been received.

Optionally, the operator may use the Mobile Data Terminal to cause an ImAlarmCancel message to the dispatch computer to notify the dispatcher of a request to terminate the alarm condition, however this message does not itself cause the alarm condition to be reset or closed, nor does it end the SlientAlarm Dialog.

The dialog ends when the dispatcher determines that the alarm was false or that the incident requiring the alarm has concluded, causes the dispatch computer to send a closing message (ImSilentAlarmClose) to the vehicle computer. The dispatch computer may reset other agency/vendor defined states related to the alarm based on the dispatcher's decision to close the alarm.

The vehicle computer upon receipt of the closing message, resets any vendor/agency defined states (e.g. Mobile Data Terminal icon changes) and closes out the alarm.

Figure 7.4 depicts the execution of the Silent Alarm dialog pattern.

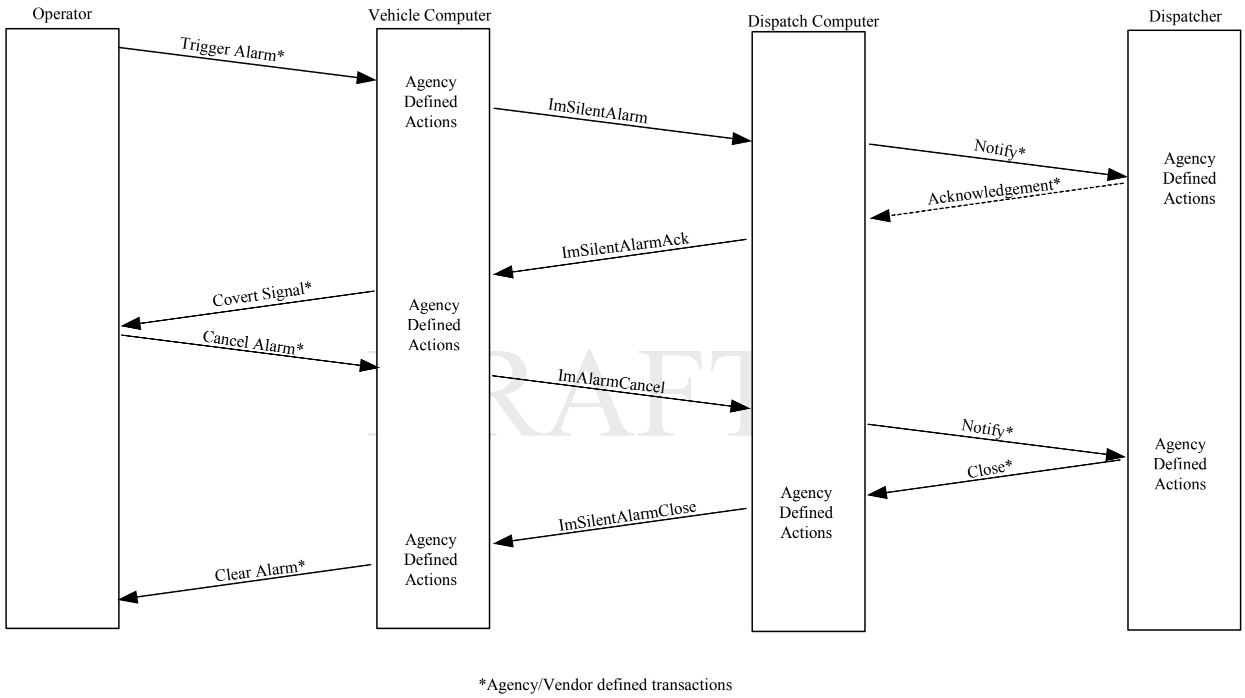


Figure 9.4 Normal Execution of the Silent Alarm Dialog

7.5 Upload Pattern

This pattern is intended to upload large files from a fixed business system to a corresponding onboard component. Files may include configuration information, software applications or other information. Uploads may be performed using a laptop computer, or a wireless LAN or other network connection. Uploads may also be performed using portable media such as flash cards or CDs (batch mode). This dialog pattern is applicable to uploads performed over a network connection or a laptop plugged into the onboard component. Normally uploads will not be performed over narrowband radio networks.

The fixed component (laptop or fixed business system) is responsible for maintaining configuration control over the files to be uploaded. The fixed component is responsible for keeping track of the correct version number(s) of the uploaded file(s) as appropriate for the onboard component. The correct current version number for an onboard component may be the same for all vehicles in the fleet or different by class of vehicle, garage-base, or by individual vehicle.

The onboard component is responsible for keeping track of the version number(s) of the upload files it has stored. Onboard components may or may not keep multiple versions of the same upload file depending on internal memory size and file management capabilities. Manufacturers of onboard components must ensure that if a file is incompletely loaded (e.g. due to network failures or loss of wireless LAN availability) that the onboard component remains usable.

The upload initiation is always performed by the mobile end by sending a CptOnboardVersionNotice message to the fixed component, however the fixed component can trigger the onboard component to initiate the upload by sending a CptForceUpload message. Note that an upload initiation may not result in an actual file transfer, as the initial exchange of messages (CptOnboardVersionNotice, and CptVerionsInfo) between the onboard and fixed components may result in a determination that the correct file versions are already on hand in the onboard component.

The onboard component will initiate an upload upon:

- Receipt of a CptForceUpload message from the fixed component.
- Receipt of a notification (via the Subscribe Wireless LAN Status Dialog) that the wireless LAN has become available and the minimum upload request interval (see below) has elapsed since the last upload initiation.
- Determination that the maximum upload request interval (see below) has expired and the wireless LAN is available.

The minimum upload request interval is a parameter that prevents the onboard components from continuously initiating uploads when in marginal wireless LAN coverage. In this situation it is possible for the Subscribe Wireless LAN Dialog to frequently indicate failures and recoveries of the wireless LAN. Instead of having the onboard component interpret each recovery as a new visit to the garage, this parameter governs how long the onboard component must wait before initiating a new upload request. This interval is overridden by receipt of a CptForceUpload message. The value of this parameter is locally defined. A recommended initial default interval is 60 minutes.

The maximum upload request interval is a parameter that prevents the onboard components from failing to initiate new uploads due to a prolonged continuous period of wireless LAN availability. If the wireless LAN is always available, the Subscribe Wireless LAN Dialog will not send event-driven notifications of wireless LAN availability resulting in the onboard components failing to obtain new uploads. If the vehicle remains in wireless LAN coverage for a period exceeding this interval, the component will initiate a new upload. The value of the maximum upload request interval is locally defined. A recommended initial default interval is 720 minutes (12 hours).

The dialog pattern executes as follows:

1. The onboard component initiates the dialog based on one of the three criteria described above and sends a CptOnboardVersion Notice message to the fixed component. This message informs the fixed component of the current version(s) of the upload files stored in the onboard component.

2. The fixed component upon receipt of the CptOnboardVersionNotice sends a CptCurrentVerionNotice to the onboard component. This message informs the onboard component of the file versions that should be stored by the onboard component, and optionally provides a list of files to delete.

3. The onboard component determines:
   A. The files that need to be loaded
   B. Whether there will be room to store the files to be loaded (allowing for space freed up by file deletions)
   C. Whether the files specified in the CptCurrentVersionNotice message are files that the onboard component should have (e.g. farebox doesn't use AVLU software files).

4. The onboard component sends a CptUploadControl message to the fixed component ending the upload dialog or requesting the first file to be uploaded. The dialog can be ended at this point due to:
   A. Invalid information in the CptCurrentVersionNotice message
   B. Insufficient storage in the onboard component
   C. All files are current

5. If a valid file upload was requested, the fixed component sends the requested file. If the file request is invalid, the fixed component sends a CptBadUploadRequest message and the dialog ends.

6. Upon receipt of the file the onboard component validates the file (manufacturer defined but including a verification of file length at a minimum), and determines whether additional (files) are needed. The onboard component sends a CptUploadControl message to the fixed component ending the dialog or requesting the next file. If a file is requested, go to step 5 above.

Figure 7.5 depicts the execution of the Upload dialog pattern.

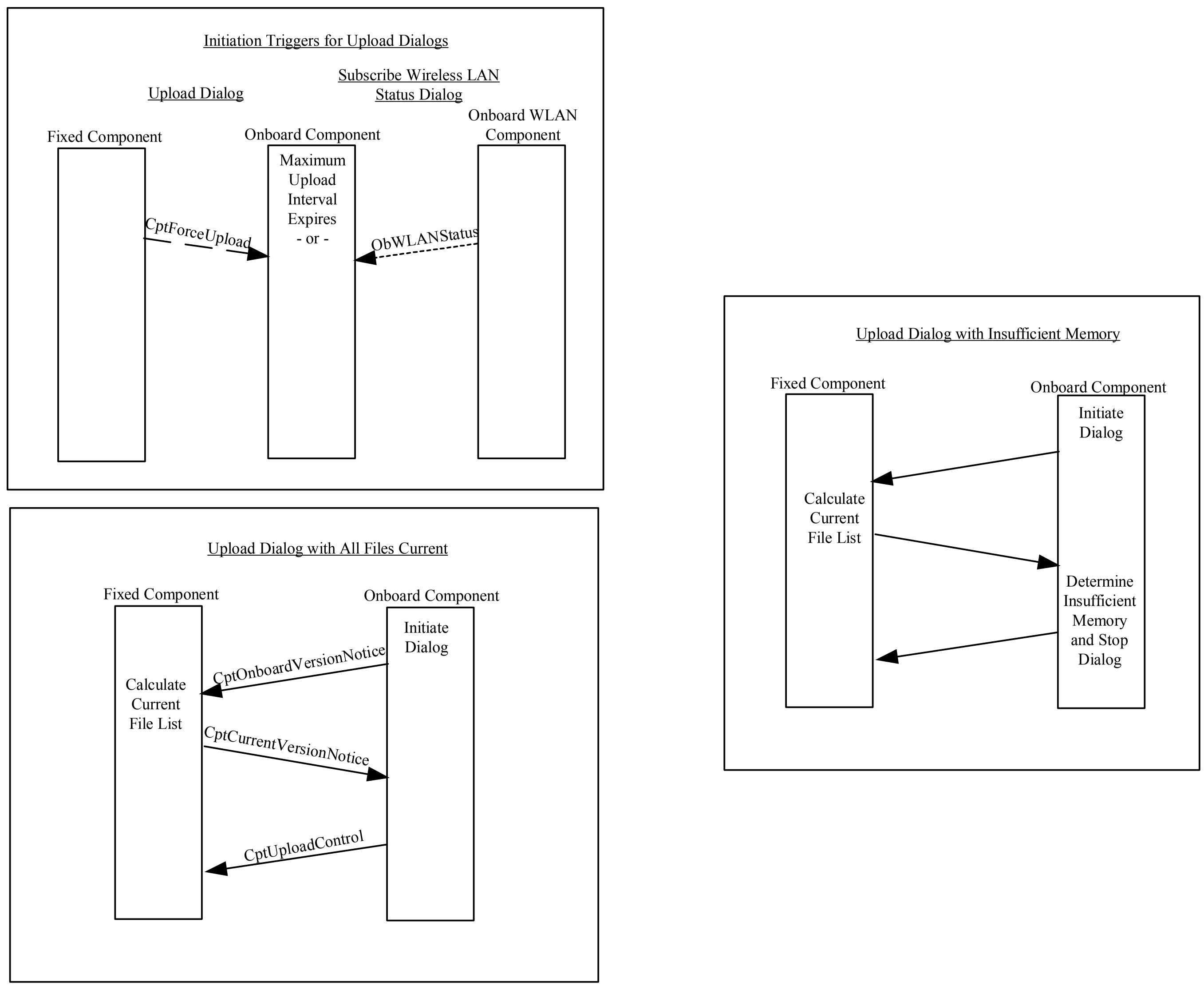


Figure 9.5 Depicts the execution of the Upload dialog pattern

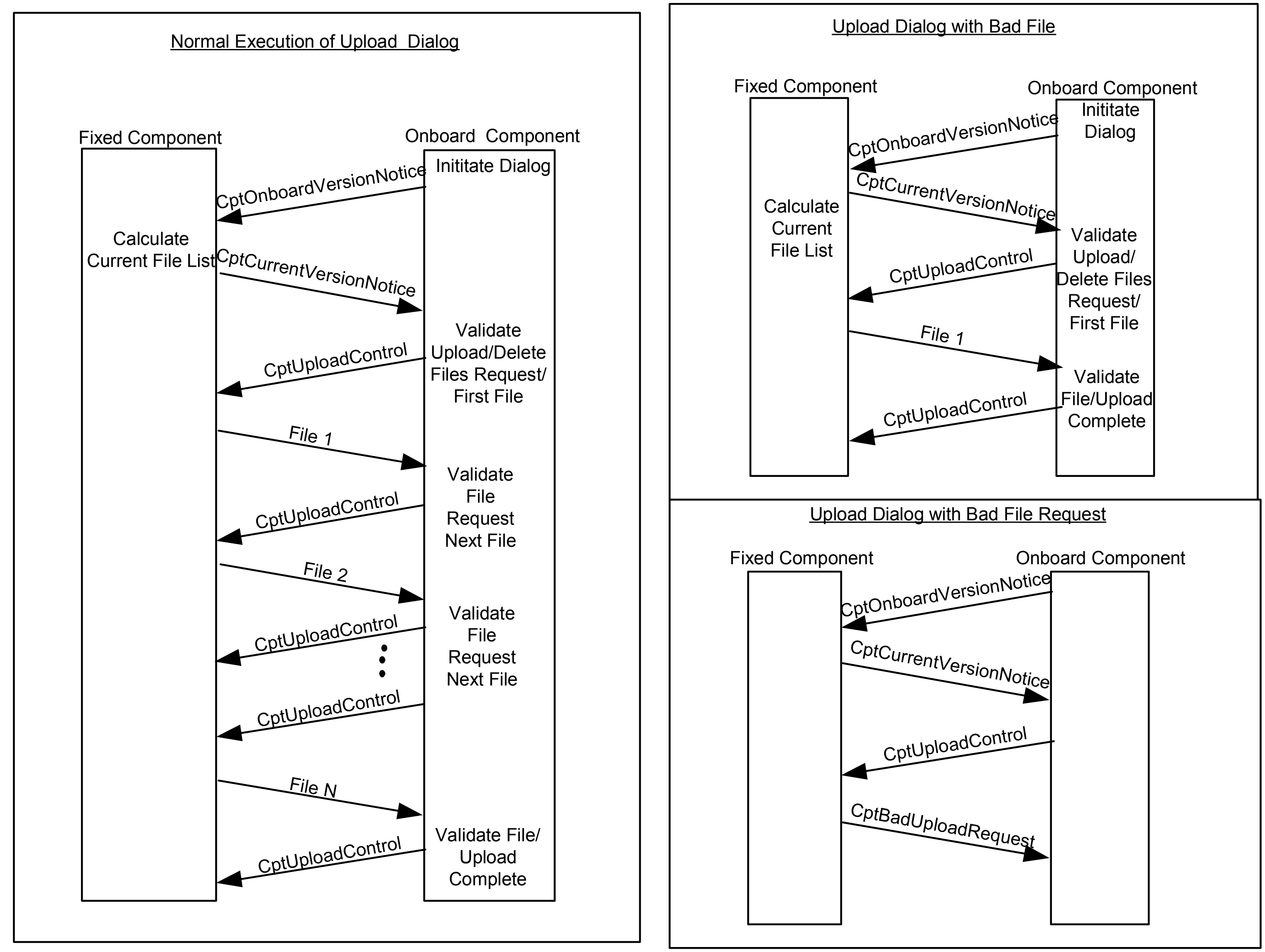


Figure 9.6 Depicts the Execution of the Download dialog pattern

7.6 Download Pattern

This pattern is intended to download large files to a fixed business system from a corresponding onboard component. Files may include configuration information, fare collection data, schedule adherence and other operator performance data or other information. Downloads may be performed using a laptop computer, or a wireless LAN or other network connection. Downloads may also be performed using portable media such as flash cards or CDs (batch mode). This dialog pattern is applicable to downloads performed over a network connection or a laptop plugged into the onboard component. Normally downloads will not be performed over narrowband radio networks.

The fixed component (laptop or fixed business system) is responsible for maintaining configuration control over the downloaded files. The onboard component is responsible for storing the files to be downloaded until the file is listed for deletion in a CptDownloadControl message or until the onboard component is forced to delete the file due to lack of storage capacity. The algorithm for determining when a file is to be deleted due to lack of storage capacity is manufacturer defined.

The download initiation is always performed by the mobile component by sending a CptFilesToDownload message to the fixed component. Note that a download initiation may not result in an actual file transfer.

The onboard component will initiate a download based upon:

- Receipt of a notification (via the Subscribe Wireless LAN Status Dialog) that the wireless LAN has become available and there are stored files waiting to be downloaded.
- Determination the wireless LAN is available, and a new file has become ready to download.
- Receipt of a CptForceDownload message from the fixed component. This message exists primarily to trigger the download to a laptop plugged into the onboard system(s).

The dialog pattern executes as follows:

1. The onboard component initiates the dialog based on one of the 3 criteria described above and sends a CptFilesToDownload message to the fixed component. This message informs the fixed component of the begin and end times for the data contained in the file, file type(s), and version number(s) of the files stored in the onboard component awaiting a download.
2. The fixed component upon receipt of the CptFilesToDownload message sends a CptDownloadControl message to the onboard component. This message informs the onboard component of the next file to download (if applicable) and any files to be deleted.
3. The onboard component determines:

   A. If an available file was requested in the CptDownloadControl message

   B. Whether any files are to be deleted without downloading them. For example the fixed component may have successfully received a download on a previous attempt, but been unable to notify the onboard component due to a loss of WLAN coverage.
4. The onboard component deletes any files specified to be deleted, and if a download was requested (that is available), initiates sending the file. If no files are requested, the dialog ends.
5. If a file is specified for deletion by the onboard component, and is not on hand in the onboard component, the deletion request is ignored.
6. If a file is specified for downloading by the fixed component and the file is not on hand, the onboard component sends a CptDownloadRequestError message to the fixed component and terminates the dialog. The onboard component may re-initiate the dialog immediately if the WLAN is still available and there are still files to download.
7. Upon receipt of a downloaded file, the fixed component sends a CptDownloadControl message to the onboard component, and the dialog goes to step 3 above. The fixed end is responsible for including the successfully downloaded file in the files to delete section of the CptDownloadControl message.

Figure 7.6 depicts the execution of the Download dialog pattern.

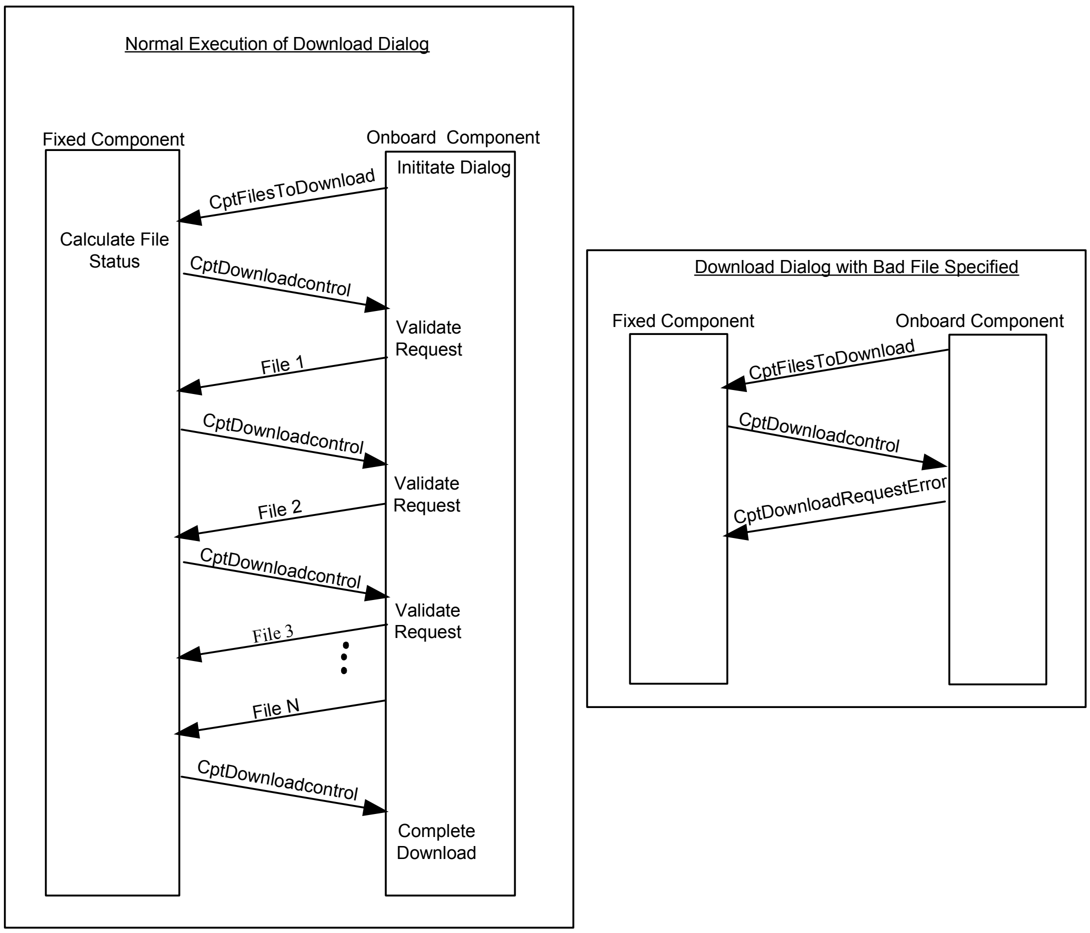


Figure 7.6 Download Dialog Pattern

7.7 Voice Radio Call Patterns

Two patterns define voice radio calls between mobiles, and central. The Operator Initiated Voice Radio Call Pattern, and the Dispatcher-Initiated Voice Radio Call Pattern. These patterns define the interactions between the VLU/MDT and the CAD/AVL System, and the annunciation system, but do not specify the radio control protocol, or the appearance or content of display screens or other operator interfaces.

These dialogs are intended to operate with a variety of one and two bus radio solutions, and thus do not assume that data messaging between the CAD/AVL system and the VLU/MDT is available while voice calls are in effect.

7.7.1 Operators Initiated Voice Radio Call Pattern

This pattern defines the sequence of events in a dialog where the vehicle operator requests a voice conversation with the dispatcher. This includes the case where the operator activated a covert microphone. The operator requests the voice call via vendor-specific mechanism such as a MDT transaction, punching a button etc. If the VLU and MDT are separate components connected by a messaging interface, and the request is made via the MDT, the ObVoiceRequest message is used to notify the VLU of the request. Other onboard architectures may not need this message.

The VLU/MDT sends a CcOperatorCallRequest message to the CAD/AVL system. The CAD/AVL system notifies the dispatcher of the request using vendor defined mechanisms. The dispatcher decides to accept or deny the call, and notifies the CAD/AVL of the result using vendor-defined mechanisms.

If the dispatcher denies the call, the CAD/AVL system sends a CcDenyCallRequest message to the VLU/MDT. The VLU/MDT informs the operator that the call request was denied and the dialog ends. If the VLU and MDT are separate components connected by a messaging interface, the ObVoiceRequestProgress message is used to notify the MDT of the denial.

If the dispatcher accepts the call, then the CAD/AVL system notifies the radio system of the call to be set up using vendor-specific mechanisms. The radio system may provide a channel number to be conveyed to the vehicle for the voice call. The CAD/AVL system sends a CcAcceptCallRequest message to the VLU/MDT. The VLU/MDT provides an indication that the call was accepted to the operator. If the VLU and MDT are separate components connected by a messaging interface, the ObVoiceRequestProgress message is used to notify the MDT of the acceptance. The VLU/MDT uses a vendor-specific means to indicate the call setup to the voice radio. If the call requires the annunciation system (e.g. covert microphone call) the VLU/MDT sends a CcAnnunciationCallSetup message to the annunciation system. At this point the voice call is established.

The operator-initiated voice call is terminated by the dispatcher who provides a vendor-specific indication to the CAD/AVL system that the call should be terminated. The CAD/AVL system uses a vendor-specific mechanism to notify the radio system that the call should be terminated. The radio system terminates the voice radio connection, providing a vendor-specific indication to the VLU/MDT that the call has ended.

The VLU/MDT notifies the operator that the call has ended. If the VLU and MDT are separate components connected by a messaging interface the ObVoiceRequestProgress message is used to notify the MDT of the call termination. If the annunciation system was included in the call, the VLU/MDT sends a CcCallTermination message to the annunciator.

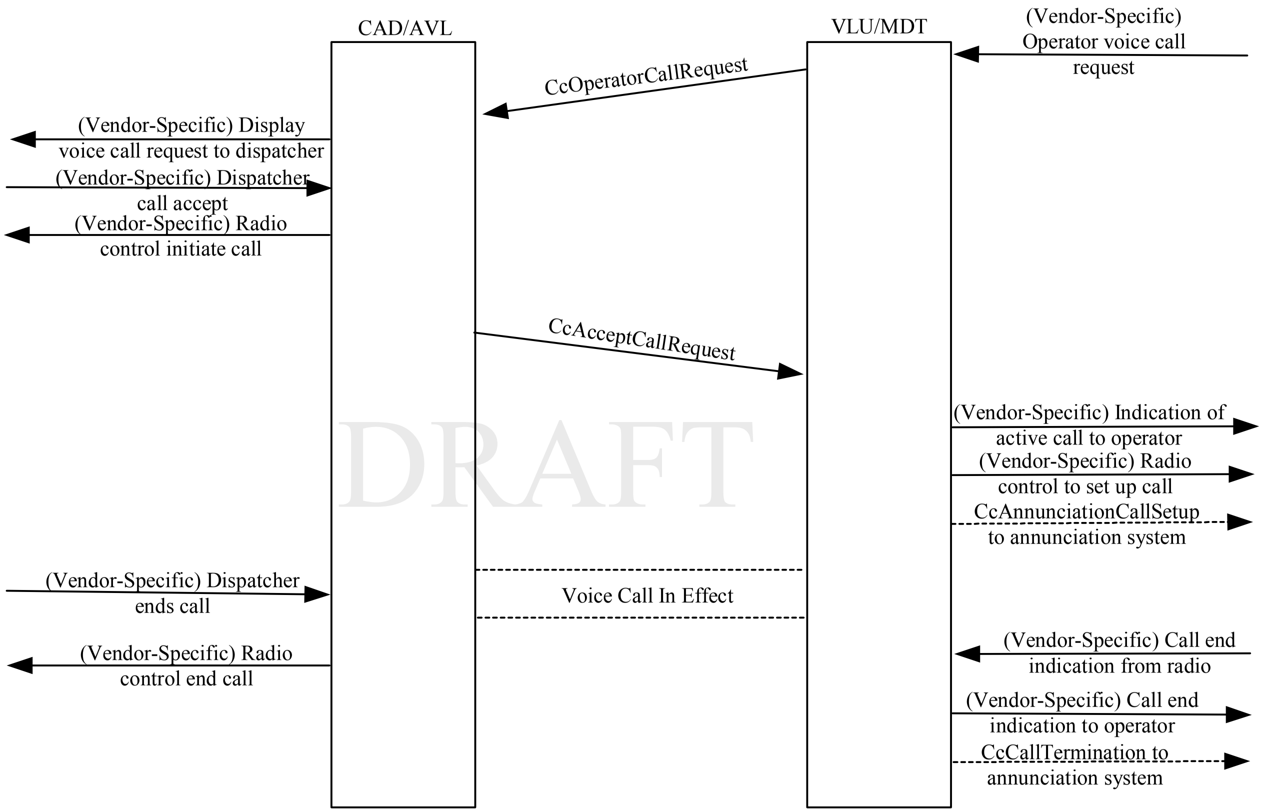


Figure 7.7.1 A Operator-Initiated Voice Radio Call Pattern (Accepted)

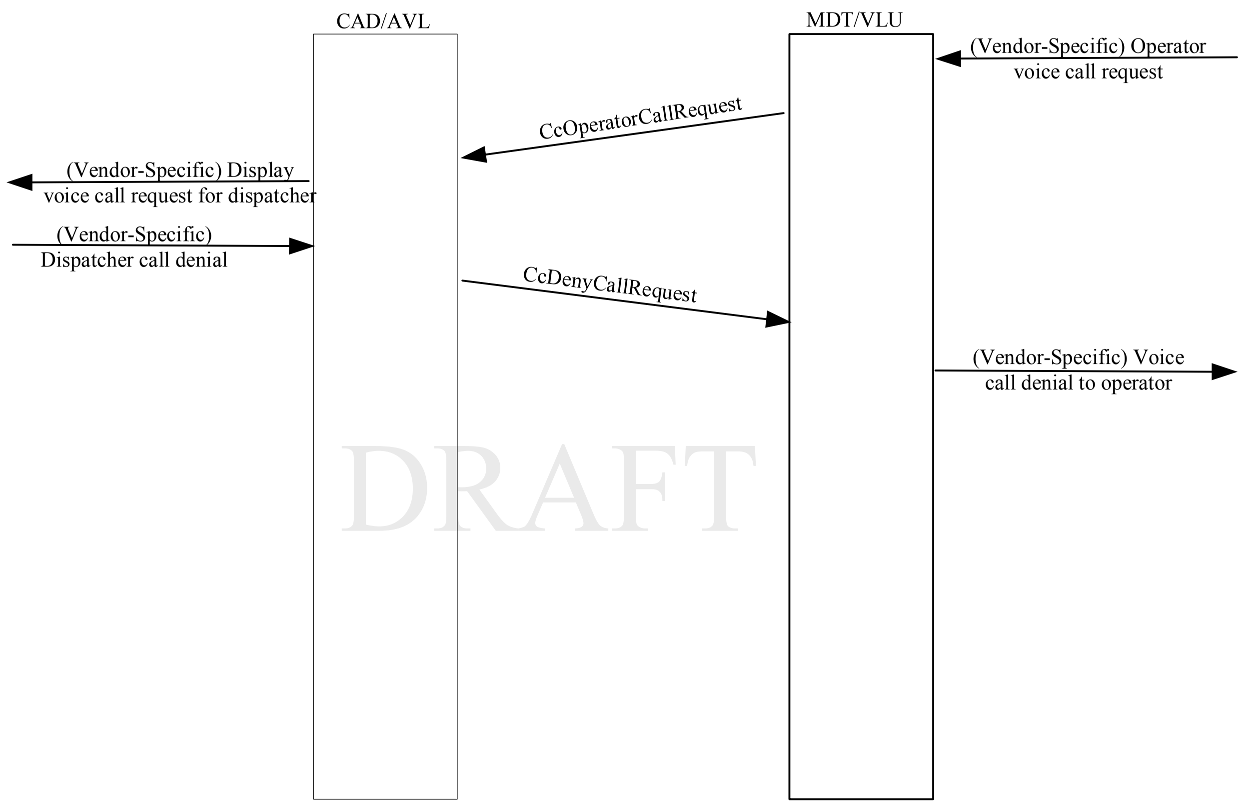


Figure 7.7.1 B Operator Initiated Voice Radio Call (Denied)

7.7.2 Dispatcher Initiated Voice Radio Call Pattern

This pattern defines a sequence of events in a dialog where the dispatcher initiates a voice conversation with the vehicle. This includes the case where the dispatcher remotely makes a voice announcement through the onboard annunciation system. The dispatcher requests the voice call via vendor-specific mechanism. The radio system may provide a channel number to be conveyed to the vehicle for the voice call. The CAD/AVL system sends a CcDispatchCallSetup message to the VLU/MDT. The VLU/MDT notifies the operator via vendor-specific mechanism of the call setup. If the VLU and MDT are separate components connected by a messaging interface, the CcNotifyIncomingCall message is used by the VLU to notify the MDT of the call setup. The VLU/MDT notifies the radio to set up the call using vendor-specific mechanisms. If the annunciation system is to be used for the call (e.g. for a dispatcher-made remote announcement). The VLU/MDT sends a CcAnnunciatorCallSetup message to the annunciator. At this point the voice call is in effect.

The call is terminated by the dispatcher similarly to the operator initiated call. The CAD/AVL System notifies the radio system to terminate the call. The onboard radio notifies the VLU/MDT of the call termination using vendor-specific mechanisms. The VLU/MDT notifies the operator of the call termination. The VLU sends a CcDispatchCallEnd message to the MDT if the VLU and MDT are separate components connected by a messaging interface. If the annunciation system was included in the call, the VLU/MDT sends a CcCallTermination message to the annunciator.

Figure 7.7.2 depicts the dispatcher-initiated voice sequence of events.

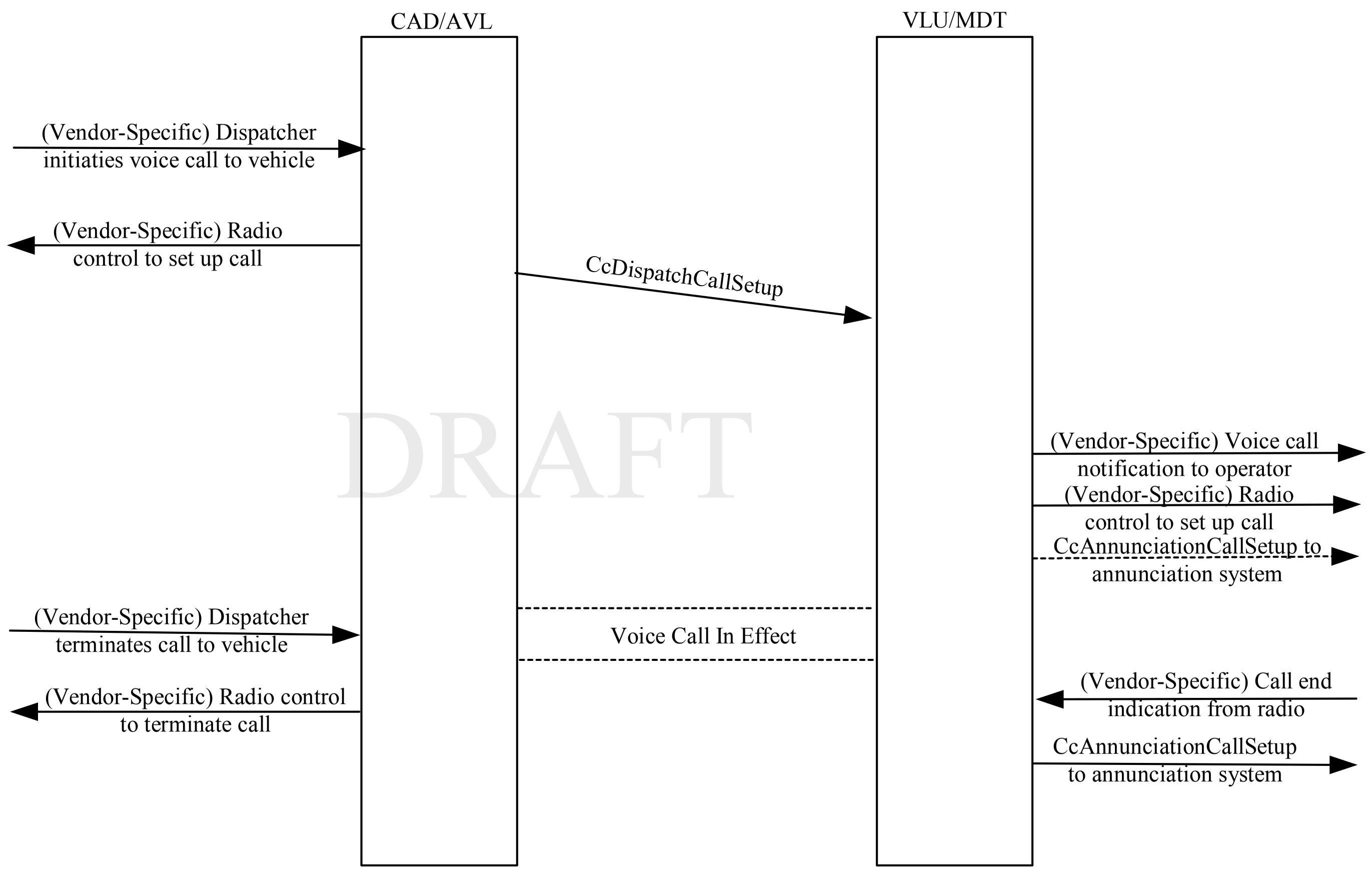


Figure 7.7.2 Dispatcher-Initiated Voice Radio Call Pattern

7.8 Signal Control & Prioritization Dialog Pattern

This pattern supports the conversations between a Priority Request Generator (PRG) that initiates requests for preferential treatment from the traffic signal system, and a Priority Request server (PRS) which processes those requests. There may be one or more intermediaries in the dialog which receive, process, and forward the messages used to request, update, status, and cancel priority requests. Intermediaries can be the Transit Control Center, the Traffic Management Center or both. This pattern is used with NTCIP 1211 Scenarios 1, 2, and 4.

In accordance with NTCIP 1211, all conversations between the PRG and PRS take place using the Simple Network Management Protocol (SNMP). Two basic SNMP processes are used in NTCIP 1211: 1) The SET procedure, which sends a value for a variable to be assigned from the PRG to the PRS. The SET procedure can also cause a variable to be allocated within the PRS if it doesn't already exist. 2) The GET procedure, which obtains the value of a variable on the PRS and returns the value to the PRG.

NTCIP 1211 defines 6 transactions for SCP as shown in Table 7.8

SCP Transactions
Table 7.8

Transaction	SNMP Type	Message	Response	Purpose
Priority Request	SET	ScpPriorityRequest	ScpPriorityRequestAck	Request a priority strategy from the PRS on behalf of a transit vehicle.
Priority Update	SET	ScpPriorityUpdate	ScpPriorityUpdateAck	Request that a previously sent priority request be modified.
Priority Status Control	SET	ScpStatusControl	ScpStatusControlAck	Prepare the PRS to receive an inquiry regarding the status of a previously sent priority request.
Priority Status Buffer	GET	ScpStatusBuffer	ScpStatusBufferRespons e	Provide the status of a previously sent priority request from the PRS to PRG.
Priority Cancel	SET	ScpPriorityCancel	ScpPriorityCancelAck	Cancel a previously send priority request that is no longer required.
Priority Clear	SET	ScpPriorityClear	ScpPriorityClearAck	Clear a completed priority request from the PRS.

The priority request is the most fundamental SCP transaction. The PRG determines a strategy to request from the PRS, and sends the request.

The priority update is intended to allow the PRG to revise the requested strategy, arrival time or other parameters related to the priority request. This transaction only occurs when changing conditions, detectable by the PRG, warrant alteration of the active request.

The priority status control is used by the PRG to condition the PRS to receive a priority status buffer. This transaction only occurs if the PRG determines it needs to obtain the current status of an active priority request.

The priority status buffer transaction is used to retrieve the priority request status from the PRS after successful execution of a priority status control.

The priority cancel is used when operational circumstances, detectable by the PRG, change such that the priority request is no longer appropriate.

The priority clear is used when the transit vehicle has cleared the intersection for which priority was requested. This transaction may (by local agreement) not be initiated by the transit vehicle to conserve communications network bandwidth. If the PRG does not generate a priority clear, the PRS will timeout and clear the request automatically.

Figure 7.8 depicts the dialog pattern for Signal Control and Prioritization. Dotted lines indicate transactions within the dialog that do not occur in every instance of the dialog's execution. Housekeeping activities by the intermediary (if one exists) generally include maintenance of a log of SCP actions, and may include maintenance of an SCP real-time display.

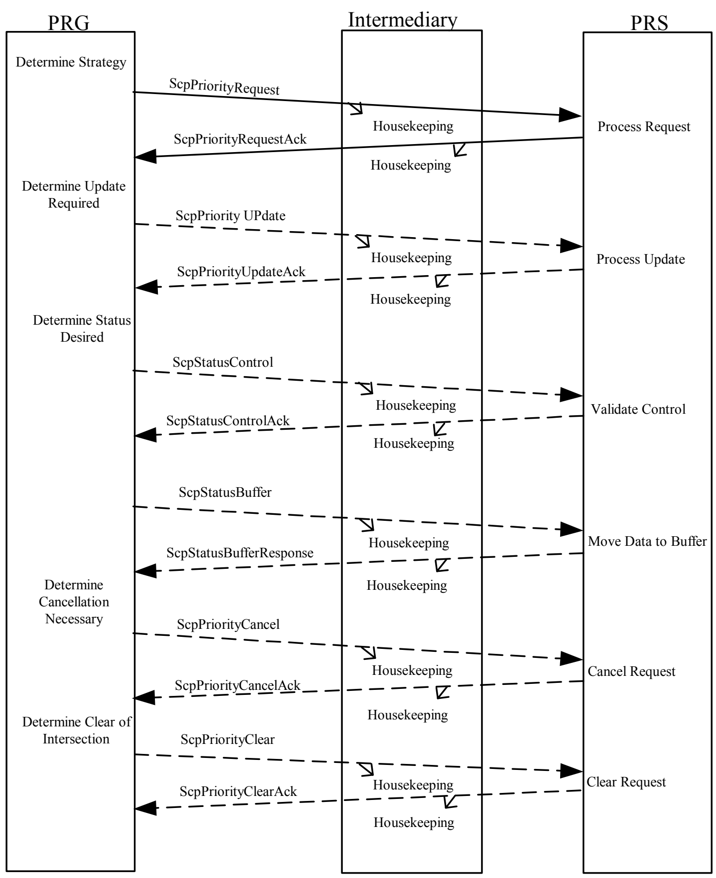


Figure 7.8
Signal Control & Prioritization Dialog Pattern

8. TCIP Data

TCIP data is defined in a three-tiered manner. Data elements are single pieces of data such as time of day, speed, vehicle identifier, or route id. Data frames are groupings of data elements (and/or smaller data frames) into larger abstractions necessary to describe more complex items. Messages are similar to data frames in that they are aggregates of data elements and data frames into a larger and more complex structure, however, messages are intended to constitute a complete understandable one-way communication rather than being a constituent of a larger object.

TCIP data was originally specified using Abstract Syntax Notation One (ASN.1). The data element, data frame, and message definitions in this document (Annexes A, B and C) include normative specifications, rules, and constraints for the data in English. These definitions also include informative ASN.1-style descriptions of the content of each item. As the TCIP data is intended to be conveyed in Extended Markup Language (XML) format, or in narrowband encoded format (see section 6.4.1.2) the normative definition of the content is included in the form of XML Schemas in Annex. To be conformant implementation must comply with the English language requirements in Annexes A, B and C, as well as the XML Schema in Annexes E.

Some TCIP dialogs need to be implemented over limited bandwidth network connections. Because XML is a verbose data representation, it is not viable over some of these narrow band links. Section 8.4.1.2 defines a narrowband encoding scheme for TCIP message for use in narrowband networks.

The table below indicates the sources of TCIP data definitions. New data definitions subsequent to the publication of the NTCIP 1400 standards are annoted as TCIP. Note that some items that were originally defined as part of the NTCIP 1400 standards have been modified in this standard to correct problems, and to increase flexibility, and to extend definitions to be consistent with new requirements.

Table 8.0 Data Definition Sources in TCIP

Table 8-1 Data Definition Sources in TCIP			
Business Area	***Data Tier***	***Range***	***Source***
CPT	Elements	cptddl – cptdd79	NTCIP 1401:2000
CPT	Frames	cpt1 – cpt20	NTCIP 1401:2000
CPT	Elements	cptdd100 – cpt999	TCIP
CPT	Frames	cpt 1000 – cpt1999	TCIP
CPT	Messages	cpt2000 – cpt2999	TCIP
IM	Elements	imdd1 – imdd86	NTCIP 1402:2000
IM	Frames	im1 – im20	NTCIP 1402:2000
IM	Elements	im100 – im999	TCIP
IM	Frames	im1000 – im1999	TCIP
IM	Messages	im2000 – im2999	TCIP

Table 8-1 **Data Definition Sources in TCIP**			
Business Area	***Data Tier***	***Range***	***Source***
PI	Elements	pidd1 – pidd70	NTCIP 1403:2000
PI	Frames	pi1 – pi15	NTCIP 1403:2000
PI	Frames	pitripreq1 – pitripreq8	NTCIP 1403:2000
PI	Elements	pidd100 – pidd999	TCIP
PI	Frames	pi1000 – pi1999	TCIP
PI	Messages	pi2000 – pi2999	TCIP
SCH	Elements	schdd1 – schdd64	NTCIP 1404:2000
SCH	Frames	sch1 – sch22	NTCIP 1404:2000
SCH	Elements	sch100 – sch999	TCIP
SCH	Frames	sch1000 – sch1999	TCIP
SCH	Messages	sch2000 – sch2999	TCIP
SP	Elements	spdd1 – spdd42	NTCIP 1405:2000
SP	Frames	spp1 – spp13	NTCIP 1405:2000
SP	Frames	spl1 – spl3	NTCIP 1405:2000
SP	Frames	sp1 – sp3	NTCIP 1405:2000
SP	Frames	spi1 – spi9	NTCIP 1405:2000
SP	Frames	spy1 – spy8	NTCIP 1405:2000
SP	Frames	spr1 – spr9	NTCIP 1405:2000
SP	Elements	spdd100 – spdd999	TCIP
SP	Frames	sp1000 – sp1999	TCIP
SP	Messages	sp 2000 – sp2999	TCIP
OB	Elements	obBusdd1 – obBusdd277	NTCIP 1406:2000 and SAE J1587
OB	Elements	obBusdd764 – obBusdd801	NTCIP 1406:2000
OB	Elements	obBusdd278 – obBusdd769	Reserved

Table 8-1 Data Definition Sources in TCIP			
Business Area	***Data Tier***	***Range***	***Source***
OB	Elements	obBusdd800-obBusdd999	TCIP
OB	Frames	obBus 770 – obBus 775	NTCIP 1406:2000
OB	Frames	obBus1000 – obBus1999	TCIP
OB	Messages	obBus2000 – obBus2999	TCIP
CC	Elements	ccdd1 – ccdd30	NTCIP 1407:2000
CC	Frames	cc1 – cc25	NTCIP 1407:2000
CC	Elements	ccdd100 – ccdd999	TCIP
CC	Frames	cc1000 – cc1999	TCIP
CC	Frames	cctemplate1 – cctemplate4	NTCIP 1407:2000
CC	Messages	cc2000-cc2999	TCIP
FC	Elements	fcdd1 – fcdd70	NTCIP 1408:2000
FC	Frames	fc1 – fc31	NTCIP 1408:2000
FC	Elements	fcdd100 – fcdd999	TCIP
FC	Frames	fc1000 – fc1999	TCIP
FC	Messages	fc2000 – fc2999	TCIP
SCP	Elements	scpdd 100-scpdd 999	TCIP
SCP	Frames	scp 1000- scp1999	TCIP
SCP	Messages	scp 2000-scp 2999	TCIP

8.1 Data Elements

TCIP data elements are based on the ASN.1 data types. These base ASN.1 types have been extended to create additional generic data types for use in creating TCIP data elements.

8.1.1 ASN.1 Data Types

TCIP uses the following ASN.1 data types as the base types to define TCIP data:

Table 8.1.1 List of ASN.1 Types

ASN.1 Type	Purpose

BOOLEAN	This base type supports True and False values. The data element CPT-Boolean uses this type.
ENUMERATED	This base type restricts a variable to a specified list of values. Data elements are defined using ENUMERATED, and a specification of the allowed values.
INTEGER	This base type is used to create several TCIP numeric subtypes. In section 6.1.2.1.1 and in 6.1.2.1.3. TCIP 1 allowed the direct use of INTEGER in data element and data frame definitions. TCIP 2 restricts these definitions to the numeric subtypes defined in this standard.
UTF8String	This base type is used to create several TCIP string subtypes in section 6.1.2.1.3. TCIP 1 allowed the direct use of UTF8String and IA5Sting in defining text string-based data elements. TCIP 2 restricts these definitions to the base types defined in this standard.
Numeric String	This base type is similar to a UTF8String, but limited to numeric values. It is used to create base types in section 6.1.2.1.3.
OCTET String	This base type is used to create base types in section 6.1.2.1.3. TCIP 1 allowed the direct use of OCTET STRING in data element and data frame definitions. TCIP 2 restricts these definitions to the base types defined in this standard

8.1.2 TCIP Subtypes of ASN.1 Data Types

The ASN.1 data types above are extended through the creation of subtypes. These subtypes and are defined in the following sections.

8.1.2.1 Subtype Definitions

This section lists the specialized subtypes based on ASN.1 that may be used in the TCIP family of standards (in addition to ASN.1 universal types).

8.1.2.1.1 Integer Subtypes

The subtypes include BYTE, UBYTE, SHORT, USHORT, LONG and UNLONG. These subtypes are defined as follows:

ASN.1 Definition	XML Simple Type	Narrowband Encoding
BYTE::=INTEGER (-128..127)	Byte	Binary signed 8 bits
UBYTE::=INTEGER (0..225)	Unsigned Byte	Binary unsigned 8 bits
SHORT::-INTEGER (-32,768 .. 32,767)	Short	Binary signed 16 bits
USHORT ::=INTEGER (0 .. 65,535)	Unsigned Short	Binary unsigned 16 bits
LONG ::= INTEGER (-2,147,483,648 .. 2,147,483,647)	Long	Binary signed 32 bits
ULONG ::=INTEGER (0 ..4,294,967,295)	Unsigned Long	Binary unsigned 32 bits

8.1.2.1.2 Date and Time Subtypes

TCIP Subtypes for defining time artifacts are TIME, DATE, DATETIME, SCHTIME, and DURATION.These Subtypes are defined as follows.

Table 8.1.2.1.2		
Date and Time Subtype Definitions		
Subtype Name	**XML Simple Type**	**Narrowband Encoding Definition**
TIME	time	ULONG constrained to the range 0…2359599999. The value is masked by digits as HHMMSS.FFF where HH represents hours, MM represents minutes, SS represents seconds and FFF represents fractions of seconds
DATE	date	ULONG constrained to the range 0…99991231. The value is masked by digits as CCYYMMDD where CC represents the century, YY represents the century, YY represents the year MM represents the month, and DD represents the day of the month.
DATE TIME	dateTime	A six octet grouping where the first 2 octets are treated as a USHORT conveying the century and year. The last 2 octets are treated as a ULONG conveying the date and time. The value is masked by digits as MMDDhhmmss where MM is the Month, DD is the date, hh is the hour mm is the minutes and ss is the seconds.
SCHTIME	duration, however only the seconds field is allowed. All other fields are to be unspecified. The specified number of seconds indicates time since midnight, negative values indicate the time before midnight (previous calendar day).	LONG interpreted as seconds since midnight. Negative values indicate seconds prior to midnight on the schedule day.
DURATION	duration	LONG interpreted as an interval in seconds.

8.1.2.1.3 Other Subtypes

Character string subtypes use a subset of Universal Multiple-Octet Coded Character Set [ISO/IEC 10646-1] representative codes. Strings are specified as UTF8String ASN.1 – Universal Type 12 except TELEPHONE which is specified as a numeric sting (ASN.1 Universal Type 18). These subtypes are identified as follows:

TELEPHONE ::= NUMERIC STRING (SIZE(0..9))
Where the 10-character string is masked as AAANNNNNNN. AAA refers to a three-digit area code; NNNNNNN refers to a seven digit number.

FOOTNOTE ::=UTF8String (SIZE(1…255))
Footnote is a memo field that is compatible with the number of characters typically allowed in spreadsheet software products.

MEMSHORT23 ::= OCTET STRING (SIZE 1..23))
Memshort 23 is a memo field of up to 23 octets.

MEMVSHORT ::= OCTET STRING (SIZE(1..255))
Memvshort is a memo field of up to 255 octets.

MEMSHORT ::=OCTET STRING (SIZE(1..2047))
Memshort is a memo field of up to 2,047 octets.

MEMLONG ::= OCTET STRING (SIZE(1..2,000,000))
Memlong is a memo field of up to 2,000,000 octets.

NAME30 ::= UTF8String (SIZE(1..30))
Name 30 is a field that contains a title or reference to a person, place or thing. The field contains up to 30 characters.

NAME2 ::=UTF8String (SIZE(1..2))
A short name form which contains up to 2 charcters.

NAME3 ::=UTF8String (SIZE (1..3))
A short name form that contains up to 3 characters.

NAME4 ::=UTF8String (SIZE (1..4))
A short name form that contains up to 4 characters.

NAME5 ::=UTF8String (SIZE (1..5))
A short name form that contains up to 5 characters.

NAME6 ::=UTF8String (SIZE (1..6))
A short name form that contains up to 6 characters.

NAME7 ::=UTF8String (SIZE (1..7))
A short name form that contains up to 7 characters.

NAME8 ::= UTF8String (SIZE(1..8))
A short name form which contains up to 8 characters.

NAME9 ::=UTF8String (SIZE (1..9))
A short name form that contains up to 9 characters.

NAME10 ::=UTF8String (SIZE (1..10))
A short name form that contains up to 10 characters.

NAME11 ::=UTF8String (SIZE (1..11))
A short name form that contains up to 11 characters.

NAME12 ::=UTF8String (SIZE (1..12))
A short name form that contains up to 12 characters.

NAME13 ::=UTF8String (SIZE (1..13))
A short name form that contains up to 13 characters.

NAME14 ::=UTF8String (SIZE (1..14))
A short name form that contains up to 14 characters.

NAME15 ::=UTF8String (SIZE (1..15))
A short name form that contains up to 15 characters.

NAME16 ::= UTF8 String (SIZE(1…16))
A short name form that contains up to 16 characters.

NAME17 ::=UTF8String (SIZE (1..17))
A short name form that contains up to 17 characters.

NAME18 ::=UTF8String (SIZE (1..18))
A short name form that contains up to 18 characters.

NAME19 ::=UTF8String (SIZE (1..19))
A short name form that contains up to 19 characters.

NAME20 ::=UTF8String (SIZE (1..20))
A short name form that contains up to 20 characters.

NAME21 ::=UTF8String (SIZE (1..21))
A short name form that contains up to 21 characters.

NAME22 ::=UTF8String (SIZE (1..22))
A short name form that contains up to 22 characters.

NAME23 ::=UTF8String (SIZE (1..23))
A short name form that contains up to 23 characters.

NAME24 ::=UTF8String (SIZE (1..24))
A short name form that contains up to 24 characters.

NAME25 ::=UTF8String (SIZE (1..25))
A short name form that contains up to 25 characters.

NAME40 ::=UTF8String (SIZE (1..40))
A long name form that contains up to 40 characters.

NAME60 ::=UTF8String (SIZE (1..60))
A long name form that contains up to 60 characters.

IDENS ::= INTEGER (0..65,535)
Where IDENS is a format for an identifier that contains 16 bits (USHORT)

IDENL ::= (0..4,294,967,295)
Where IDENL is a format for an identifier that contains 32 bits (ULONG)

8.1.3 TCIP Data Element Definition

TCIP numbers the defined data elements within each business area. Data elements defined in the original standards are identified by the business area abbreviation (e.g. cpt, sch) and a number. Data elements added by APTA in this document are identified by the business area abbreviation and a number higher than the numbers used in the original TCIP standards, see Table 6-1.

TCIP data elements are based on the ASN.1 types defined in 6.1.1 above and the TCIP subtypes defined in 6.1.2 above. For example ASN.1 defines UTF8String (tag number 12). Section 6.1.2 defines NAME30 derived from

UTF8String as NAME30::=UTF8String (size(1…30)). TCIP defines data elements based on the NAME30 subtype, as in the definition of cptdd 18

CPT-EmployeeFirstName::=NAME30.

Finally the data element definition is used to create an XML Schema. The corresponding schema entry for this element is

```
<xsd:simpleType name="CPT-EmployeeFirstName" >
<xsd:restriction base ="NAME30"/>
</xsd:simpleType>
```

8.2 TCIP Data Frames

Data frames are groupings of data elements (and/or smaller data frames) into larger abstractions necessary to describe more complex items. For instance a data frame describing an employee might include name, employee number, date of employment and/or other information. Depending on the context, the description of an employee might require more or less information. As an example, the employee's date of hire might be important to a personnel action, but not relevant to an assignment as an operator for a particular bus run. Consequently similar data frames may be created for use in different contexts, and/or data frames may be created with optional data fields that can be used in some contexts and not others. In some cases two optional fields may be incompatible, requiring explanatory material in the data frame usage area.

The NTCIP 1400 series of standards defined "messages" which could either be interpreted as complete messages or as data frames to be included in other messages. All of these original "messages" are included in this document as data frames, and are used where appropriate within the messages defined in Annex C. Data frames defined in the original NTCIP 1400 series of standards are identified by the business area abbreviation and a number below 1000. Data frames added subsequent to the initial NTCIP 1400 series standards are identified by the business area abbreviation and a number 1000 or higher. See Table 8-1 for sources of data frames by business area and number. Figure 8.2 illustrates the definition of data frames in TCIP

Commentary:

The "messages" originally defined in the NTCIP 1400 series specifications were designed in a fashion that allowed them to either be interpreted as message fragments or as complete messages. This resulted in ambiguity in determining what constitutes a complete transmission. The rearrangement of the original "messages" into the data frames tier, and the creation of a new separate message tier resolves this ambiguity. TCIP Dialogs exchange messages as defined in Annex C. Data frames are used exclusively as building blocks which (along with data elements) are used to build messages.

Example Data Frame Definition in ASN.1

Included Data Frames

```
PiAmenity::=SEQUENCE{
amenityID          PI-AmenityID,
stopAttribute      CPT-StoppoinAttribute,
location           SpPointClass    OPTIONAL,
name               PI-AmenityName,
stopID             CPT-StoppointID,
address            SpAddressPoint
}
```

Included Data Frames

Example Data Frame Definition in XML

```
<xsd:complexType name="PiAmenity" >
    <xsd:sequence>
      <xsd:element name="amenityID" type="PI-AmenityID" />
      <xsd:element name="stopAttribute" type="CPT-StopPointAttribute" />
      <xsd:element name="location" type="SpPointclass" minOccurs="0"/>
      <xsd:element name="name" type="PI-AmenityName"
      <xsd:element name="stopID" type="CPT-StopPointID" />
            <xsd:element name="SpAddressPoint" />
    </xsd:sequence>
</xsd:complexType>
```

8.3 TCIP Messages

Messages are aggregates of data elements and data frames into a larger and more complex structure. Messages are intended to constitute a complete understandable one-way communication. Messages are not intended to be aggregated into larger data objects for communications. Sequences of messages are defined in the dialogs. These sequences create complete conversations between entities providing two (or more) way communications. TCIP messages are identified by business area abbreviation and a number 2000 or higher. Figure 8.3 illustrates the definition of TCIP messages.

TCIP Message Definition in ASN.1

Included Data Elements

```
PiNearestStopListSub ::= SEQUENCE{
subscriptionInfo        CPTSubscriptionHeader,
request                 PiNearestStopRequest,
includeDistance         PI-Distance       OPTIONAL
}
```

Included Data Frames

TCIP Message Definition in XML

```
<xsd:complexType name="PiNearestStopListSub" >
    <xsd:sequence>
      <xsd:element name="subscriptionInfo" type="CptSubscriptionHeader" />
      <xsd:element name="request" type="PiNearestStopRequest" />
      <xsd:element name="includeDistance" type="PI-Distance"minOccurs="0"/>
    </xsd:sequence>
</xsd:complexType>
```

8.4 TCIP Messaging Over Narrowband Radio Links

8.4.1 TCIP Fixed-Mobile/Field Communications

TCIP Supports two types of communications between mobile and field elements including PTVs and fixed end business systems:

- Bulk Transfer Communications, and
- Narrow Band Communications

8.4.1.1 Bulk Transfer Communications

Bulk Transfer Communications support file transfers between onboard/field equipment and fixed-end based business systems. Bulk Transfers may be implemented using batch techniques to transfer files on removable media such as smart cards, compact disks, or diskettes. Vendor-defined procedures govern the process of transferring the information to and from the removable medium while the format of the data in the files is defined by TCIP (in XML). Bulk Transfers may also be used to transfer files over high-speed wireless technologies such as IEEE 802.11 or DSRC.

Bulk Transfers over high-speed wireless links ("wireless LANs") use dialogs defined within the appropriate business area to transfer data between field/mobile systems and fixed end systems. In general the initiation of these dialogs is from the field/mobile end. Since PTVs and other mobiles spend significant periods of time in locations without access to high-speed wireless support, the high-speed data communications system is responsible for notifying the mobile equipment when the link is, and is not, available using the SubscribeWirelessLANStatus dialog. This allows subscribing components or applications onboard the vehicle to initiate uploads or downloads with their corresponding fixed side business systems while the wireless link is up, independent of the agency or vehicle's architecture.

8.4.1.2 Narrow Band Radio Communications

Limited capacity public or private data networks are widely used by transit agencies to track and interact with public transit vehicles, supervisory vehicles, maintenance vehicles and other agency mobile assets. To accommodate communications over such networks, TCIP specifies narrowband messages – "Narrowband TCIP" encoding. Narrowband TCIP messages are designed to be compact to make efficient use of the scarce capacity of narrow band links. Consequently, Narrowband TCIP messages are specified as binary bit patterns rather than as eXtended Markup Language (XML) schemas. The impact of this is that Narrowband TCIP messages cannot be used with XML parsers, but require software at the sending and receiving ends that "understand" the Narrowband TCIP binary format.

8.4.1.2.1 Narrowband TCIP Message Order of Octet and Bit Transmission

Since Narrowband messages are defined in binary, the order of bit and octet transmission must be defined. The following rules govern the transmission of Narrowband TCIP Messages:

- Data elements within a Narrowband TCIP message are transmitted in the order listed in the message definition
- String and text fields are transmitted character by character using ASCII 8 bits no parity. Characters are transmitted in the order normally read. Bits within a character are transmitted least significant bit first.
- Numeric fields are transmitted most significant octet first within a number, least significant bit first within an octet.
- Bit fields are transmitted as are numerics, that is most significant octet first, with the least significant bit first in each octet.

8.4.1.2.2 Narrowband TCIP Message Encoding Restrictions

There are no restrictions on the TCIP messages that can be encoded using the Narrowband TCIP encoding rules – all TCIP messages can be encoded into Narrowband TCIP messages. While there are no restrictions on which TCIP messages can be encoded using the Narrowband encoding rules, it is expected that only those interface with bona fide bandwidth constraints will use the non-xml version of TCIP.

8.4.1.2.3 Narrowband TCIP Message Encoding Technique

The Narrowband TCIP Message encoding mechanism is designed to achieve significant compression of TCIP messages when compared to XML encoding, while using a relatively simple algorithm to control software complexity, and at the same time maintaining the capability to take advantage of key constructs such as data frames, optional fields, and sequences of elements/frames.

Each Narrowband TCIP Message consists of a message header and a message body as shown in Figure 6.4.1.2.3-1.

> Commentary: The Narrowband TCIP encoding scheme does not attempt to optimize data compression of TCIP messages, but strives to maintain a balance between data compression, ease of use and future growth considerations. Consequently, message header information is not packed to the maximum extent possible.

<u>Commentary:</u> Decoding of Narrowband TCIP encoded messages requires that the decoding entity has access to the detailed structure information for each encoded message. This information includes parameters such as the specific elements and order of elements within a message, OPTIONAL designations, enumerated codes, message lengths, etc.

8.4.2.1.3.1.1 Narrowband Message Header

The first five octets in the message header consist of an 8-bit business area ID, a 16-bit message number, and a 16-bit TCIP version number field.

Following the message identifier octets, an additional 1 to 15 octets, depending on the total number of both OPTIONAL and non-OPTIONAL fields in the TCIP Message, are included in the Narrowband TCIP Message header. These octets provide an entry-present-map for the message by identifying which of the entries defined for the message are present in this instance of the message. The low order bit of the map represents the first entry in the message, and each successively higher order bit represents a subsequent message entry as shown in Figure 1. A bit set to zero indicates that a field is not present, and a bit set to one indicates that the field is present. If an entry is not specified as OPTIONAL or a choice alternative in the message definition, then it is an error to set the bit corresponding to that field to zero. Bits representing entries beyond the last entry specified in the message definition (if any) shall be set to zero.

8.4.1.2.3.1.2 Narrowband Message Body

The Message Body that follows the Narrowband TCIP Message Header consists of a series of Data Fields, one for each entry in the message that is present, per the entry-present-map in the Message Header. If an entry is OPTIONAL and not present, no octets representing that missing entry are provided in the Message Body. If a CHOICE constructor is used, each choice value is treated as a separate entry, although only one entry present-bit may be set to true for the listed choice alternative entries.

There are 5 different types of Data Fields that can be included in the Message Body:

- Type 1: Fixed Length Element
- Type 2: Variable Length Element
- Type 3: ENUMERATED Element
- Type 4: Data Frame / Message
- Type 5: SEQUENCE OF constructor

The detailed encoding rules for each of these Data Field types are provided in Figures 2 through 7.

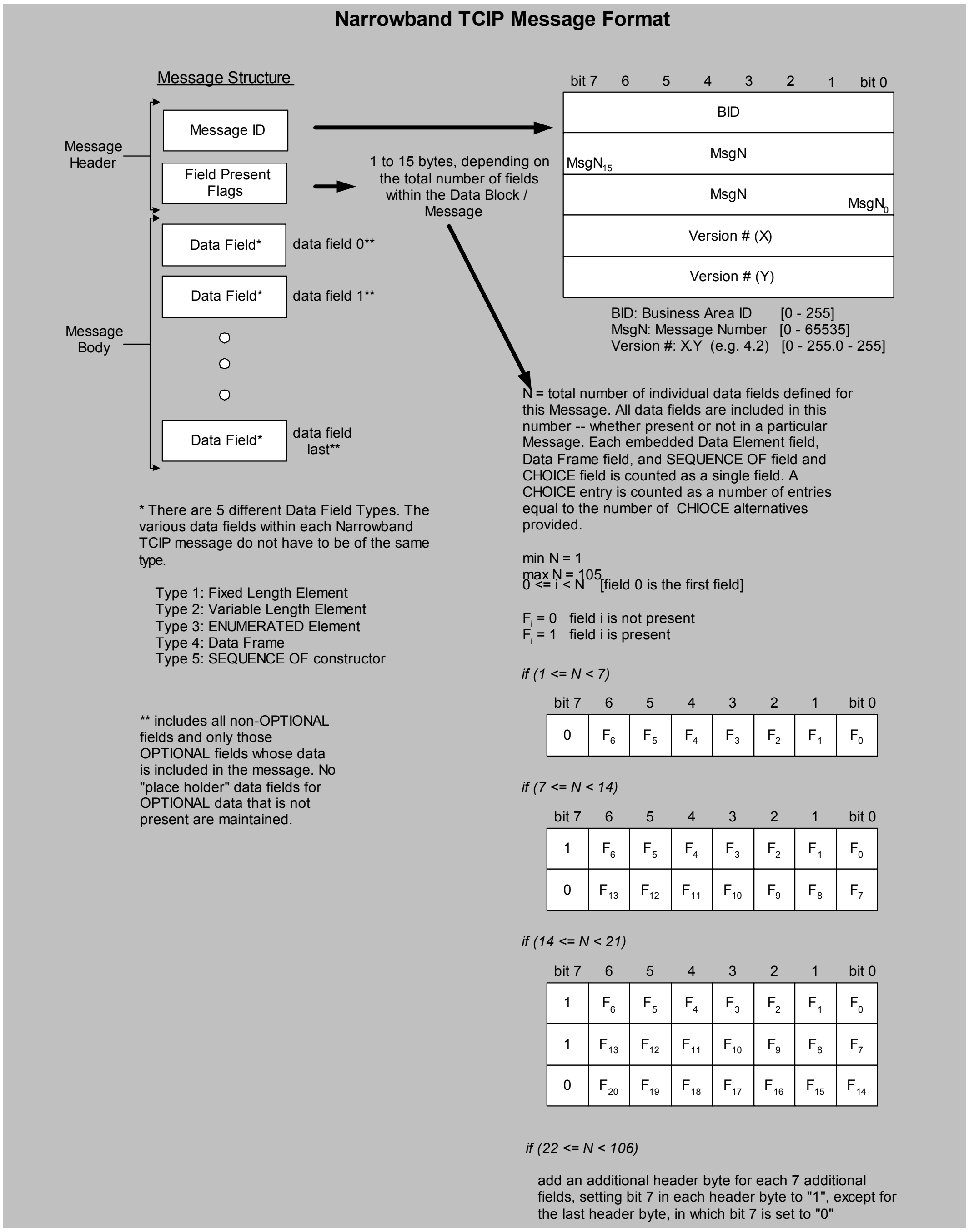


Figure 8.4.1.2.3.1.2: Narrowband TCIP Message Format

8.4.1.2.3.2 Fixed Length Data Elements

Fixed Length data elements are data elements with the following base types and subtypes:

BOOLEAN – one byte field
BYTE – one byte field
DATE – four byte field

DATETIME – six byte field
DURTIME – four byte field
IDENL – four byte field
IDENS – two byte field
LONG – four byte field
SCHTIME – four byte field
SHORT – two byte field
TELEPHONE – ten byte field
TIME – four byte field
UBYTE – one byte field
ULONG – four byte field
USHORT – two byte field

The fixed length fields are placed in the message most to least significant byte. In the case of the TELEPHONE type the numeric characters are placed in the message in the order the digits are dialed.

Figure 8.4.1.2.3.2 depicts the implementation of the fixed length data element type.

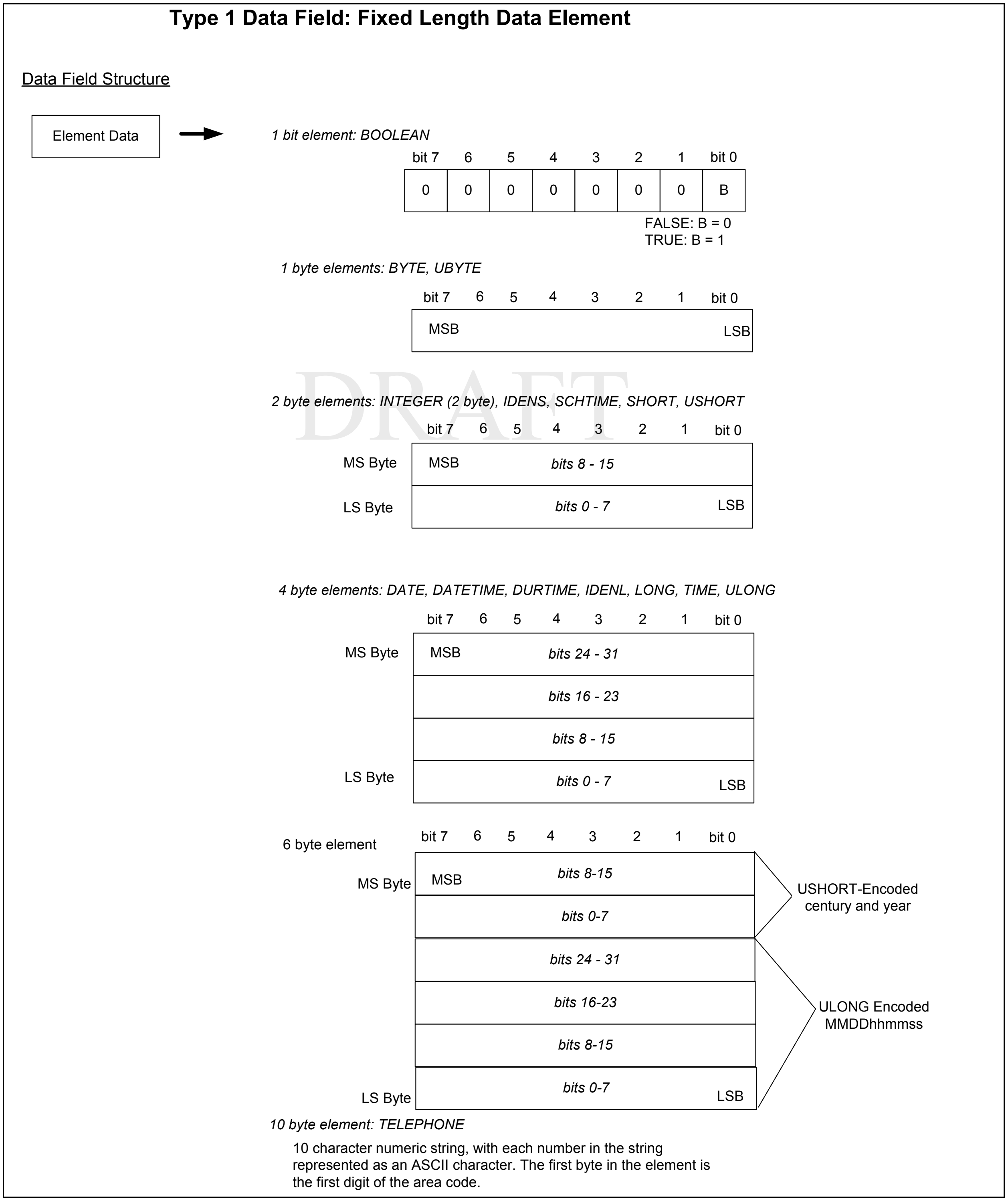


Figure 8.4.1.2.3.2: Narrowband TCIP Fixed Length Element Type

8.4.1.2.3.3 Variable Length Data Elements

Variable length data elements are data elements with the following base types and sub types:

FOOTNOTE	-up to 255 characters
MEMLONG	-up to 2,000,000 octets
MEMSHORT23	-up to 23 octets
MEMSHORT	-up to 2047 octets
MEMVSHORT	-up to 255 octets
NAME30	-up to 30 characters
NAME2	-up to 2 characters
NAME3	-up to 3 characters
NAME4	-up to 4 characters
NAME5	-up to 5 characters
NAME6	-up to 6 characters
NAME7	-up to 7 characters
NAME8	-up to 8 characters
NAME9	-up to 9 characters
NAME10	-up to10 characters
NAME11	-up to 11 characters
NAME12	-up to 12 characters
NAME13	-up to 13 characters
NAME14	-up to 14 characters
NAME15	-up to 15 characters
NAME16	-up to 16 characters
NAME17	-up to 17 characters
NAME18	-up to 18 characters
NAME20	-up to 20 characters
NAME21	-up to 21 characters
NAME22	-up to 22 characters
NAME23	-up to 23 characters
NAME24	-up to 24 characters
NAME25	-up to 25 characters
NAME40	-up to 40 characters
NAME60	-up to 60 characters

A variable length field is represented in the message by an element length subfield, and an element data subfield. The element length subfield stores the length information in the lower 7 bits of each octet, and uses the most significant bit (7 bit) as a continuation bit. The element length subfield is a maximum of 3 octets long allowing it to support up to 2,097,152 length element data fields.

The element data subfield follows the element length subfield and contains as many octets as are specified by the element data subfield. For UTF8String types, the characters in the string are stowed in the element data subfield in the order normally read. For OCTET STRING types, the octets are stored in from first to last, in the message, in the same order as their appearance in the OCTET STRING.

Figure 8.4.1.2.3.3 depicts the implementation of variable length data elements.

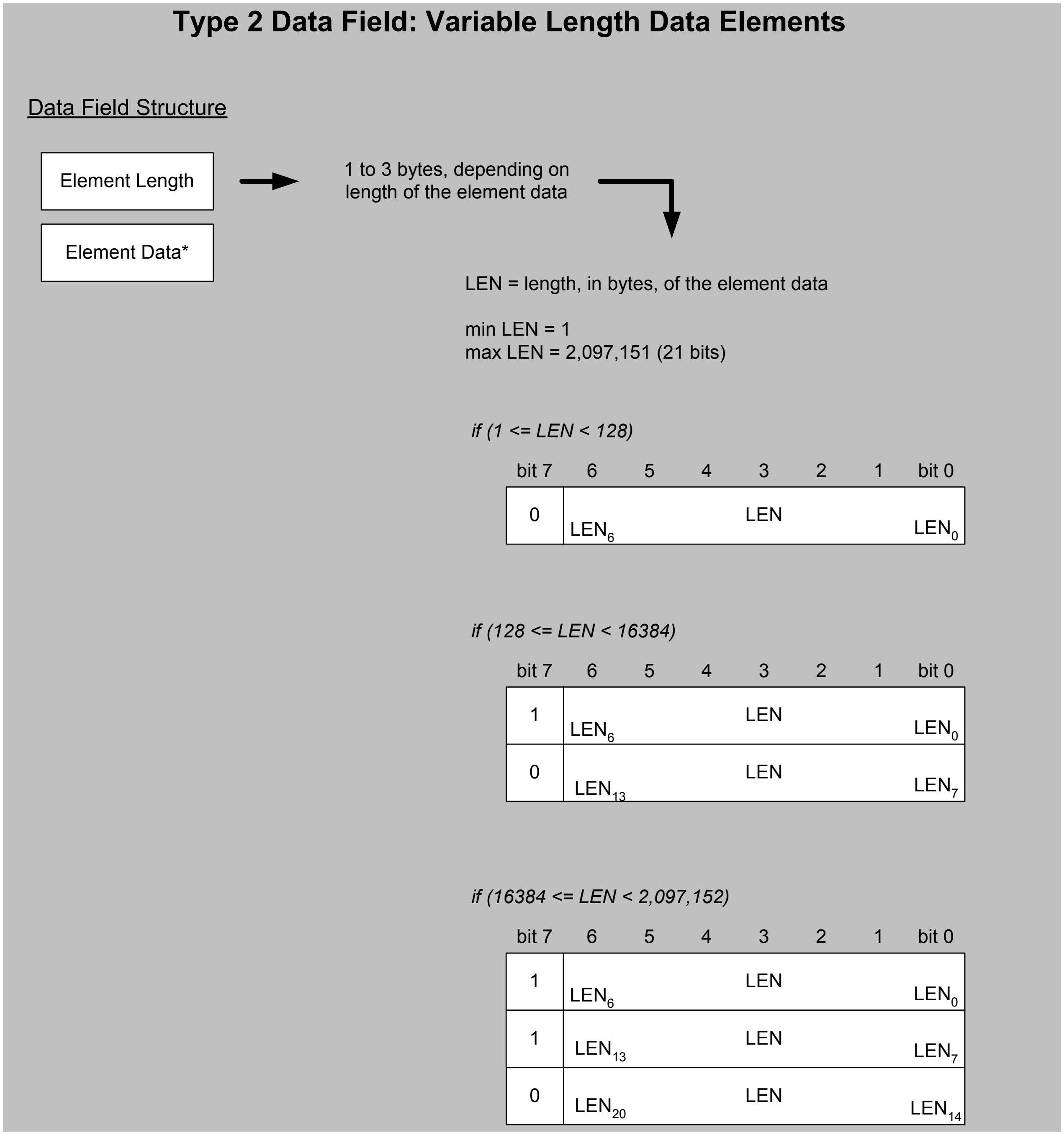


Figure 8.4.1.2.3.3: Narrowband TCIP Variable Length Data Element Type

8.4.1.2.3.4 Enumerated Data Elements

Enumerated data fields are signified by the use of the ENUMERATED keyword in their ASN.1 definition. Some TCIP enumerated data types define specific numeric index values for each token value listed for the data element. In those cases the index value used to identify the token shall be the number assigned to that token in the data element definition. For any enumerated element without specified index values corresponding to each token, the index value of the first specified token in the data element definition shall be zero. Each subsequently defined token value shall have a corresponding index value incremented by one.

For example, if a data element definition specified allowed values of A,B,C,D and E, and no corresponding index values were specified, then the index values would be:
A=0
B=1
C=2
D=3
E=4

And values 5-255 would be invalid.

If, on the other hand the data element definition specified that the values were:
A=1
B=2
C=3
D=4
E=5
--6-10 reserved
--11-20 local use
--21-200 reserved

Then the values 0, 6-10, and 21-255 would be invalid. The values 1-5 would be used as index values as indicated in the data element definition. The values 11-20 would be valid if locally defined, and would have the locally defined significance. If the values 11-20, or any subset of those values were not locally defined, those undefined values would be invalid.

Figure 8.4.1.2.3.4 depicts the implementation of Enumerated Data Elements.

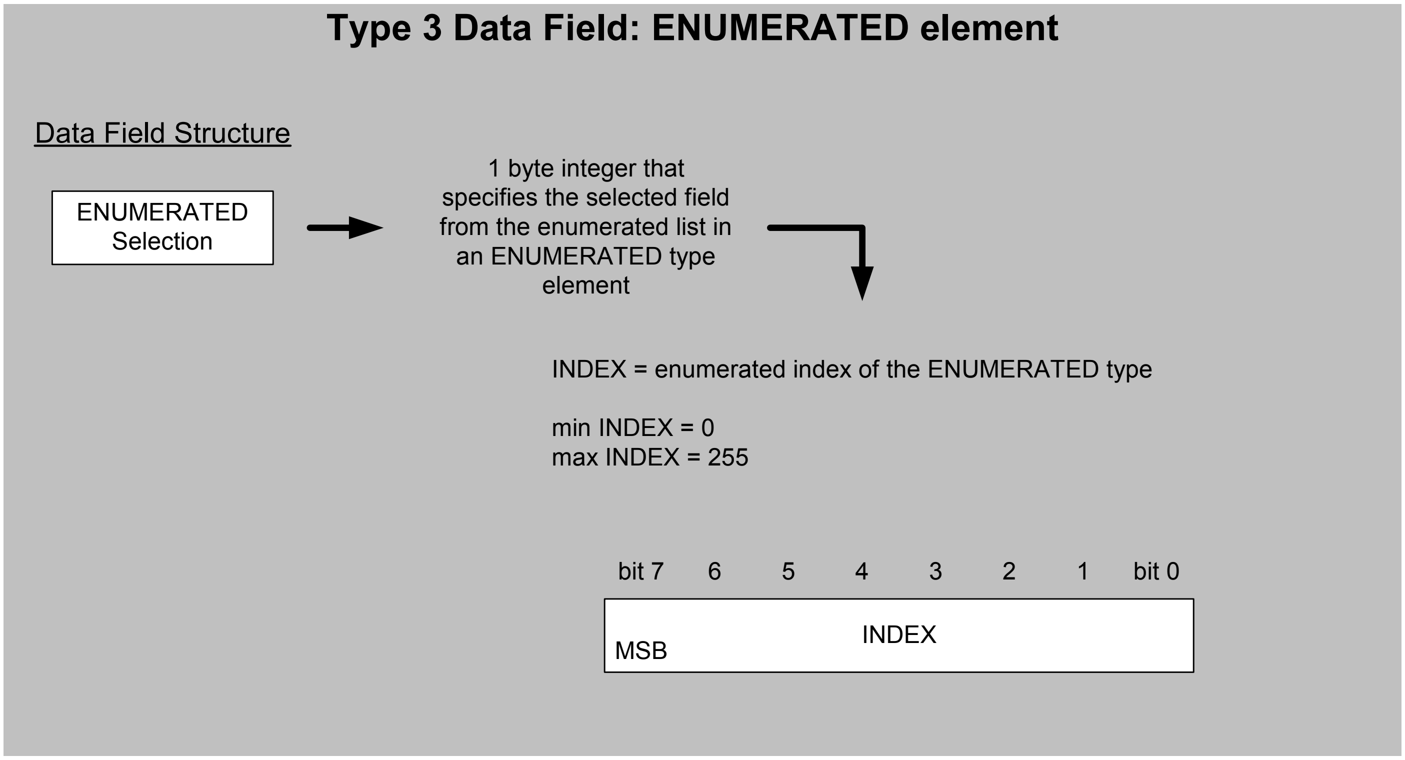


Figure 8.4.1.2.3.4: Narrowband TCIP ENUMERATED Element Type

8.4.1.2.3.5 Data Frames

Data frame encoding is performed similarly to the message encoding, but without the message identifier. 1-15 bytes encode the entry-present-map for the frame. CHOICE fields are encoded as if the CHOICE were not present. A separate bit defines the presence or absence of the CHOICE alternative. A bit set to zero indicates that the field is not present, and a bit set to one indicates that the field is present. If an entry is not specified as OPTIONAL or is a CHOICE alternative in the message definition, then it is an error to set the corresponding bit to zero. Bits representing entries beyond the last entry specified in the data frame definition (if any) shall be set to zero.

The same 5 types of data fields allowed in a message body are allowed in a data frame body.

Type 4 Data Field: Data Frame

Data Field Structure

Field Present Flags → 1 to 15 bytes, depending on the total number of fields within the Data Frame

Data Field* — data field 0**

Data Field* — data field 1**

○
○
○

Data Field* — data field last**

* can be any Type Data Field. The Data Fields do not have to be of the same type

** includes all non-OPTIONAL fields and only those OPTIONAL fields whose data is included in the message. No "place holder" data fields for OPTIONAL data that is not present are maintained.

N = total number of individual data fields defined for this Data Frame. All data fields are included in this number -- whether present or not in a particular Data Frame. Each embedded Element field, Data Frame field, and SEQUENCE OF field is counted as a single field. A CHOICE field is counted as a number of entries equal to the numer of CHOICE alternatives provided.

min N = 1
max N = 105

0 <= i < N [field 0 is the first field]

$F_i = 0$ field i is not present
$F_i = 1$ field i is present

if (1 <= N < 7)

bit 7	6	5	4	3	2	1	bit 0
0	F_6	F_5	F_4	F_3	F_2	F_1	F_0

if (7 <= N < 14)

bit 7	6	5	4	3	2	1	bit 0
1	F_6	F_5	F_4	F_3	F_2	F_1	F_0
0	F_{13}	F_{12}	F_{11}	F_{10}	F_9	F_8	F_7

if (14 <= N < 21)

bit 7	6	5	4	3	2	1	bit 0
1	F_6	F_5	F_4	F_3	F_2	F_1	F_0
1	F_{13}	F_{12}	F_{11}	F_{10}	F_9	F_8	F_7
0	F_{20}	F_{19}	F_{18}	F_{17}	F_{16}	F_{15}	F_{14}

if (22 <= N < 106)

add an additional header byte for each 7 additional fields, setting bit 7 in each header byte to "1", except for the last header byte, in which bit 7 is set to "0"

Figure 8.4.1.2.3.5: Narrowband TCIP Data Frame Type

8.4.1.2.3.6 SEQUENCE OF Constructor

The sequence of constructor provides the encoding for a sequence of data frames or data elements inside a message or data frame. The data field begins with a 16-bit count field that defines the number of items in the sequence. The sequence of items follows the count. All items in the sequence must be of the same type.

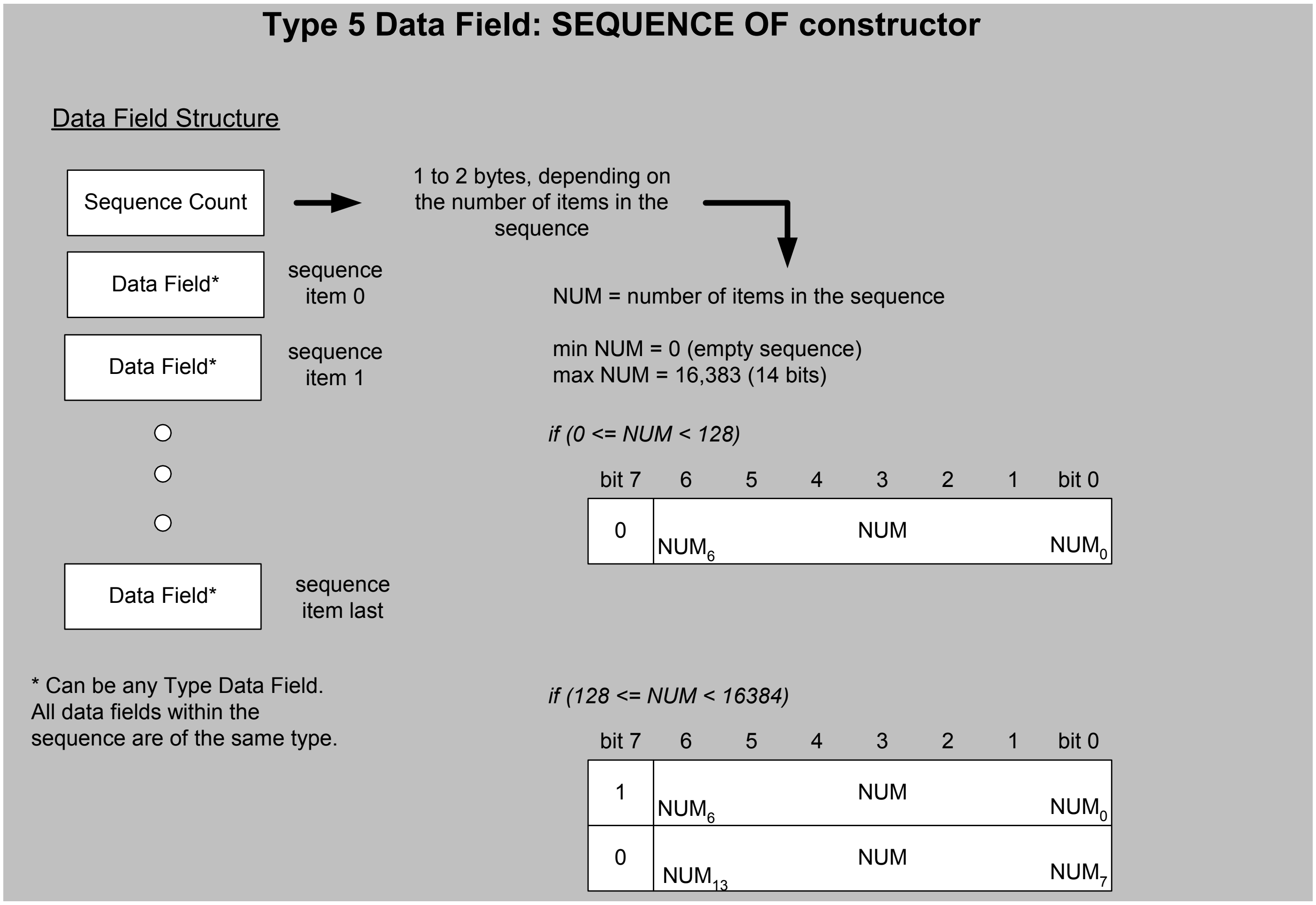


Figure 8.4.1.2.3.6: Narrowband TCIP SEQUENCE OF Type

9. TCIP Usage in RFPs

TCIP provides the building blocks that agencies can use to create agency architectures and RFPs. Dialogs can be used to specify the information flows in agency systems. Typical architectures will employ some legacy data flows and some TCIP dialog exchanges between systems. Agency architectures may include a migration path showing how the agency will acquire, evolve, and replace its business systems, and how the functionality of the systems and the interfaces between them are affected at each stage of the evolution.

Each building block depends in some way on the blocks upon which it rests. At the top of the stack of blocks is the agency RFP. This depends on the dialogs to be implemented by the new system(s), the agency's terms and conditions, the legacy systems (to be replaced or to remain in place and interact with new system(s)). At the bottom of the stack are TCIP's lowest level objects and the key agency starting points-legacy systems, and terms and conditions which reflect agency business policies.

Figure 9.0 depicts the build up from TCIP components to agency RFP's.

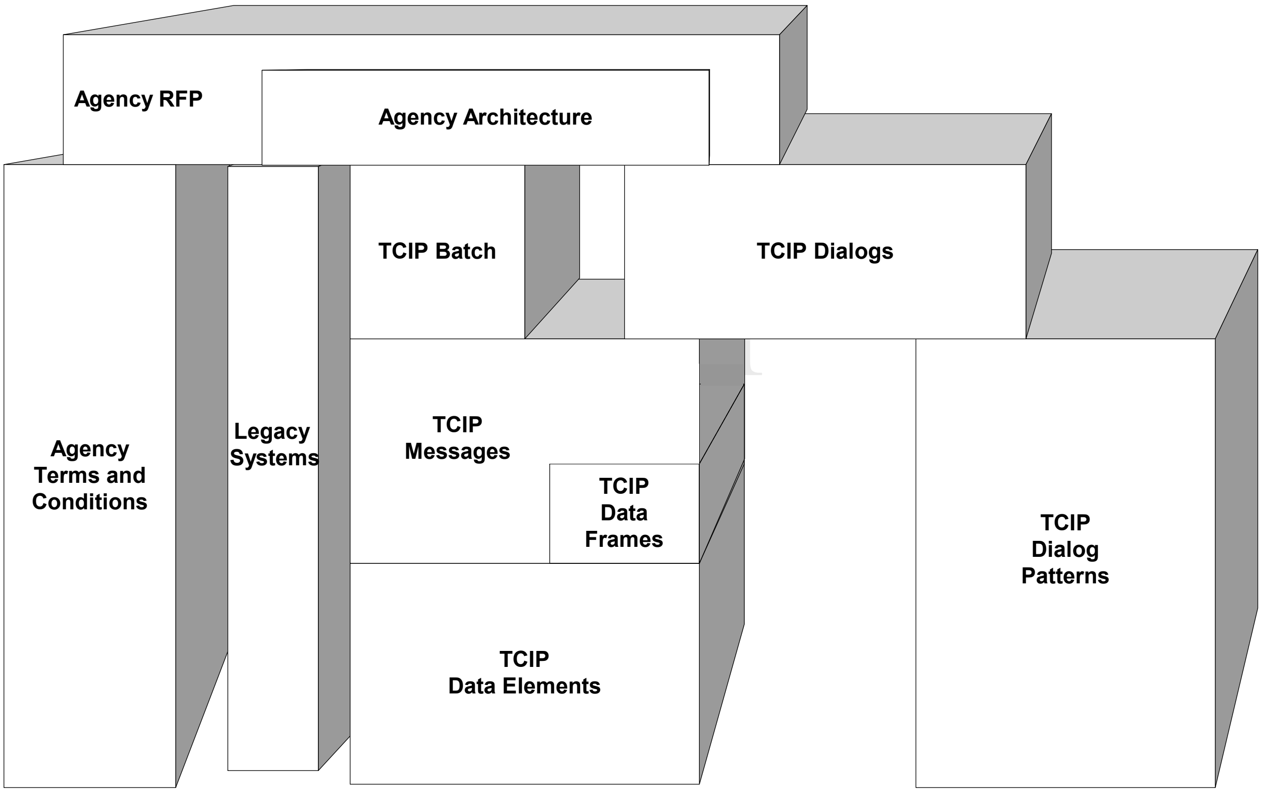
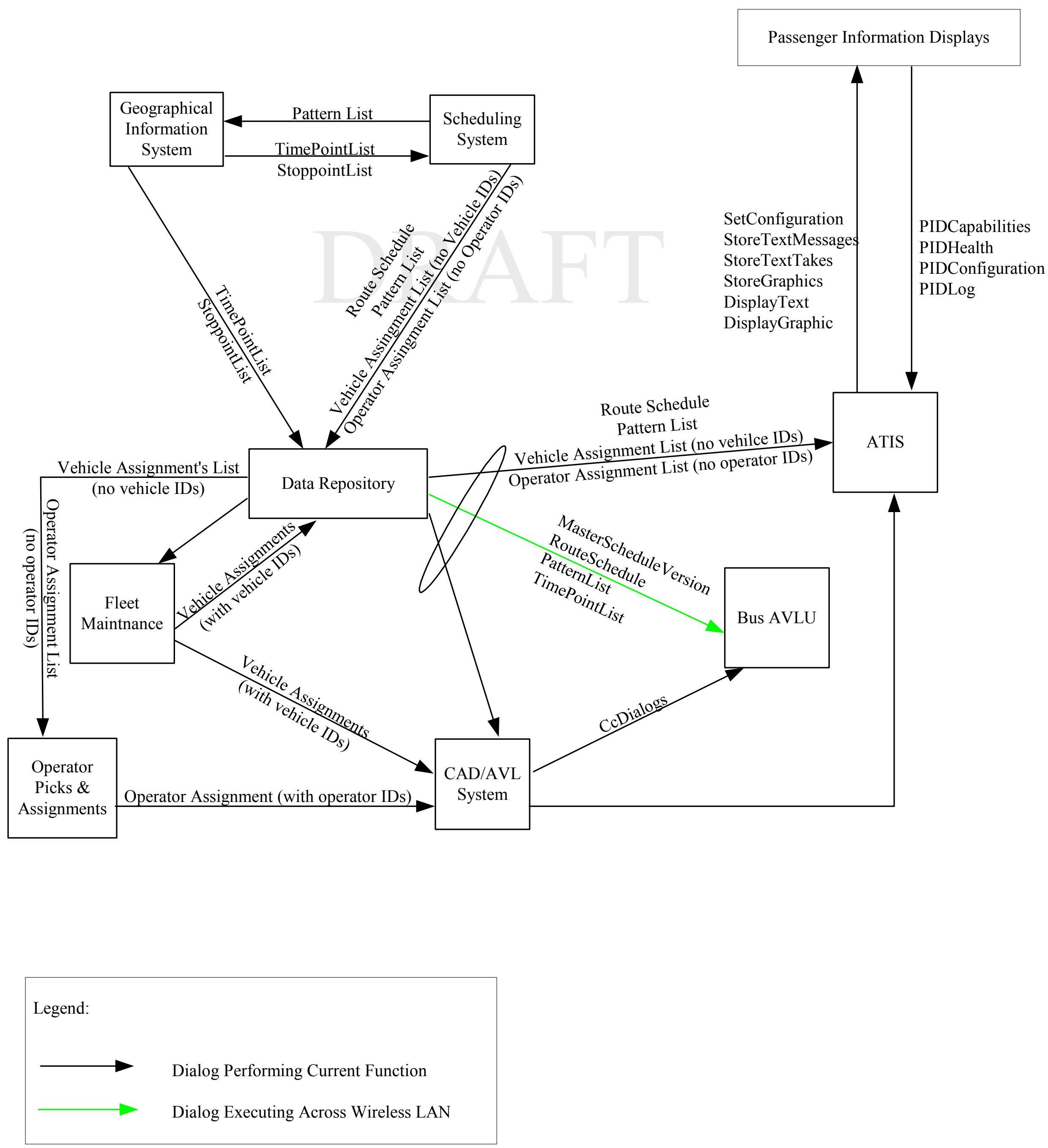


Figure 9.0 TCIP Provides the Building Blocks for Agency Architectures and RFPs

9.1 Example Agency Architecture Excerpt

Figure 9.1 depicts a simplified agency architecture for an agency that separately maintains a scheduling/runcutting system, a Geographical Information System (GIS), a data repository and other business systems. The depicted information flows are each supported by a dialog with a corresponding name (e.g. the "TimePointList" information flow is supported by the "Subscribe Time Point List" dialog).

Figure 9.1 Using TCIP Dialogs to Implement Information Flows in an Agency Architecture



9.2 Creating an RFP

The technical specifications in the agency's RFP for a business system are developed by:

- Determining the interfaces between business systems to be implemented and identifying the appropriate TCIP for those interfaces.
- Mapping the agency's legacy data and architecture to TCIP data and dialogs.
- Requiring the vendors to include the appropriate side of each of the dialogs specified in the architecture with the products.
- Specifying what optional and configuration items associated with each dialog the agency wants. For example some dialogs allow either a one time query or an event-based subscription to be used for an information flow giving the agency the flexibility to determine which type best suits their environment.
- Specifying what optional fields the agency wants included in the data flows. For example a message may identify a bus route by identity number and provide an option to also convey the route's name. The agency would determine whether to require the name to be sent with the identity number.
- Specifying the communications methods and media (protocols) to be used to transfer messages between the system being procured and other systems.

<There might be some more things to cover here-would like comments>

D.9 Signal Control & Prioritization Dialogs

TCIP Dialog Definition Page 1

Dialog Name: Upload SCP Data

Business Area: SCP

Dialog Pattern: Upload

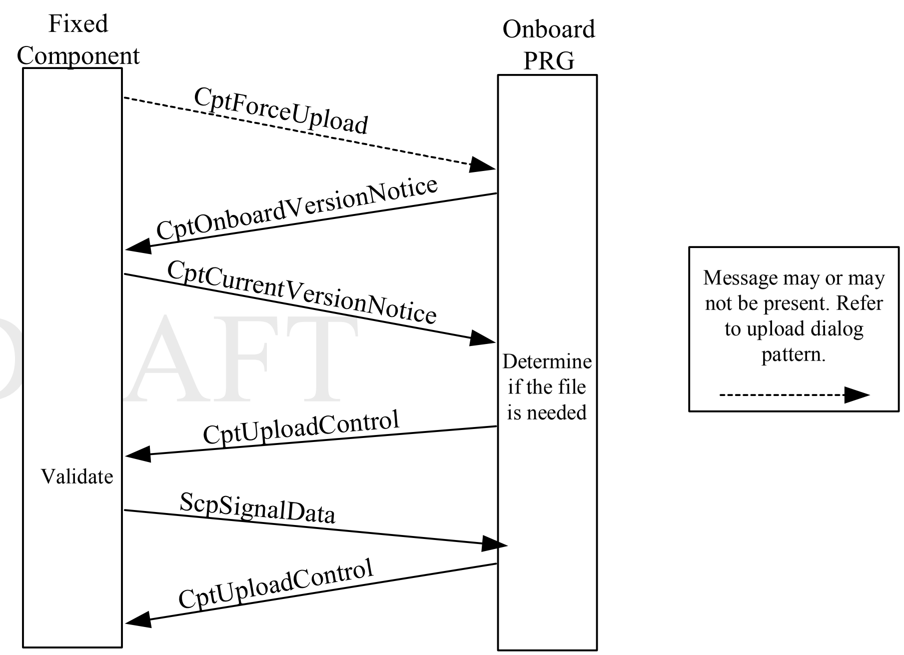
Purpose: Provide information to a Signal Control & Prioritization – Priority Request Generator (PRG) onboard the bus. This information is required by the PRG to generate and manage priority requests in NTCIP 1211 Scenarios 1 and 4.

Assumptions: The fixed component can be the control center, or a transit data repository.

Narrative:

1) The PRG triggers the dialog (based on the state of the wireless LAN, internal timers, and/or CptForceUPload message from the fixed component), by initiating a CptOnboardVersionNotice/CptCurrentVersioinNotice exchange with the fixed component. The PRG determines if a new SCP data load is required.
2) If the SCP data on hand is current, the PRG sends a CptUploadControl message ending the dialog.
3) If the SCP data on hand is not current, the PRG sends a CptUploadControl message requesting the current version.
4) The fixed component validates the request and terminate the dialog with a CptBadUploadRequest message (if the file request is invalid), or sends the current ScpSignalData message to the PRG.
5) The PRG receives and validates the ScpSignalData message and sends a CptUploadControl message terminating the dialog.

Message Sequence Diagram Page 2



Normal Execution of the "Upload SCP Data" Dialog

TCIP Dialog Definition Page 3

Dialog Name: Upload SCP Data

Business Area: Scp

Dialog Pattern: Upload

Message Name	Message Identifier	Role
CptForceUpload	Cpt 2011	Used by the fixed component to force the PRG to initiate the upload.
CptOnboardVersionNotice	Cpt 2010	Used by the PRG to notify the fixed component of the ScpSignalData version number on hand.
CptCurrentVersionNotice	Cpt 2009	Used by the fixed component to notify the PRG of the current version of the ScpSignalData.
CptUploadControl	Cpt 2007	PRG controls the upload process with this message.
ScpSignalData	Scp 2000	Conveys the data necessary for the PRG to function.
CptBadUploadRequest	Cpt 2008	Fixed component aborts the dialog with an error notice to the PRG with this message.

Notes:

TCIP Dialog Definition Page 1

Dialog Name: Scp Priority Request Scenario 1

Business Area: SCP

Dialog Pattern: Signal Control & Prioritization

Purpose: This dialog defines the generation and processing of priority requests by an onboard vehicle Priority Request Generator (PRG), with the transit Control Center as an intermediary according to NTCIP 1211 Scenario #1.

Assumptions:

1) The PRG has already received SCP data via the "Upload SCP Data" dialog.
2) The PRG has access to data resident in the vehicle logic unit (VLU)such as vehicle location, speed, bearing, schedule, and passenger count (if so equipped).

Narrative:

1) The PRG determines that the vehicle is approaching an SCP-equipped Scenario #1 intersection, and that the criteria are met to initiate a priority request. The PRG sends a ScpPriorityRequest message to the Priority Request Server (PRS) via the Control Center (CC).
   a. The control center performs local housekeeping actions and forwards the priority request to the PRS.
   b. The PRS validates the request and sends a ScpPriorityRequestAck to the PRG via the CC.
   c. The CC performs local housekeeping and forwards the acknowledgement to the PRG.
   d. If the acknowledgement contains an error indication, the dialog ends, otherwise the PRG may optionally initiate any of items 2-5 below.
2) The PRG determines that it requires a status update on the priority request. The PRG sends a ScpStatusControl to the PRS via the CC.
   a. The CC performs local housekeeping actions and forwards the status control to the PRS.
   b. The PRS validates the status control and returns a ScpStatusControlAck message to the PRG via the CC.
   c. The CC performs local housekeeping and forwards the acknowledgment to the PRG.
   d. If the acknowledgement contains no error indication, the PRG sends a ScpStatusBuffer to the PRS via the CC.
   e. The CC performs local housekeeping and forwards the status buffer to the PRS.
   f. The PRS fills the status buffer and returns it to the PRG via the CC as a ScpStatusBufferResponse message.
   g. The CC performs local housekeeping and forwards the status buffer response to the PRG.

3) The PRG determines that the priority request needs to be changed as a result of changed arrival time at the intersection and/or changes to the criteria matching. The PRG sends a ScpPriorityUpdate message to the PRS via the CC.

   a. The CC performs local housekeeping actions and forwards the priority update to the PRS.

   b. The PRS processes the update and sends a ScpPriorityUpdateAck message to the PRG via the CC.

   c. The CC performs local housekeeping actions and forwards the acknowledgement to the PRG.

4) The PRG determines that the priority request needs to be canceled as the vehicle is not progressing towards the stop bar as planned, or has deviated from its route, or the operator manually initiates a cancel (if so equipped), or the vehicle is taken out of service, or experiences another locally defined state change requiring a cancellation. The PRG sends a ScpPriorityCancel to the PRS via the CC.

   a. The CC performs local housekeeping actions and forwards the priority cancel to the PRS.

   b. The PRS processes the cancellation and sends a ScpPriorityCancelAck to the PRG via the CC.

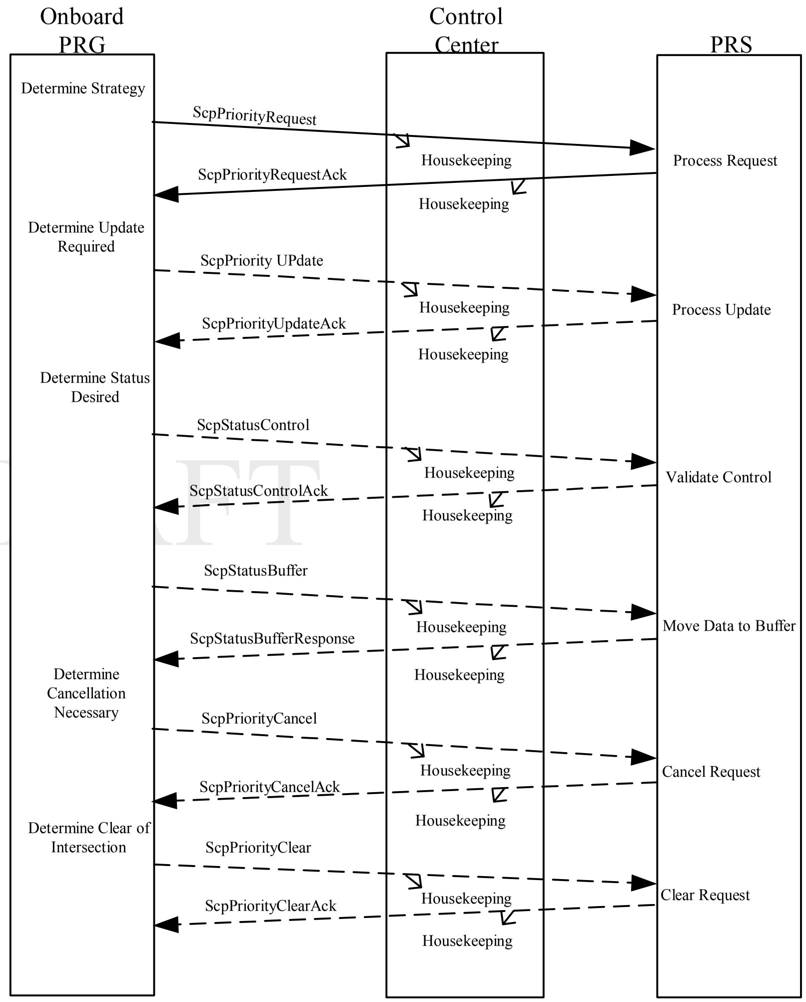
   c. The CC performs local housekeeping actions and forward the acknowledgment to the PRG and the dialog ends.

5) The PRG determines that the vehicle is clear of the intersection, and the intersection requires explicit clears. The PRG sends a ScpPriorityClear message to the PRS via the CC.

   a. The CC performs local housekeeping actions and forwards the priority clear to the PRS.

   b. The PRS clears the request and sends a ScpPriorityClearAck to the PRG via the CC.

   c. The CC performs local housekeeping actions and forwards the acknowledgement to the PRG and the dialog ends.



Normal Execution of "Scp Priority Request Scenario 1" Dialog

TCIP Dialog Definition Page 4

Dialog Name: Scp Priority Request Scenario 1

Business Area: SCP

Dialog Pattern: Signal Control & Prioritization

Message Name	Message Identifier	Role
Scp Priority Request	2011	Request a priority strategy from the PRS.
ScpPriorityRequestAck	2010	Acknowledge the priority request.
ScpPriorityUpdate	2009	Request that a previous priority request be modified.
ScpPriorityUpdateAck	2008	Acknowledge the priority update.
ScpStatusControl	2007	Request the PRS to prepare to provide status for a precious priority request.
ScpStatusControlAck	2006	Acknowledge the status control
ScpStatusBuffer	2004	Provide a buffer to obtain a priority request status
ScpStatusBufferResponse	2005	Return the status of the priority request to the PRG
ScpPriorityCancel	2002	Request the PRS to cancel a previous priority request
ScpPriorityCancelAck	2003	Acknowledge the priority cancellation
ScpPriority Clear	2012	Request the PRS to clear a completed priority request
ScpPriorityClearAck	2001	Acknowledge the priority clear.

Notes:

1) The PRG to PRS messages contain optional fields to convey intersection identifiers consistent with the Institute of Traffic Engineers Traffic Management Data Dictionary. These fields maybe used by the CC to direct the ongoing message to the proper PRS, however the CC is then responsible for removing the optional fields from the message, and forwarding a message to the PRS that complies with NTCIP 1211. Similarly, an agency may elect to use TCIP narrowband encoding, or XML encoding between the PRG and the CC, however, the CC is then responsible for format the outgoing messages to the PRS consistent with NTCIP 1211, and to reformat incoming messages from the PRS consistent with its local policy. Reformatting, if required, should be considered part of the CC housekeeping function.

2) If the vehicle clears the intersection and the PRG is not configured to generate a priority clear to the PRS, the dialog ends from the vehicle PRG view point, and ends from the CC and PRS view points after a local timeout.

3) The criteria for determining when a priority update, status update, or priority cancel should be initiated are agency/vendor defined.

TCIP Dialog Definition Page 1

Dialog Name: SCP Priority Request Scenario 2

Business Area: SCP

Dialog Pattern: Signal Control & Prioritization

Purpose: This dialog defines the generation and processing of priority requests by a transit control center based Priority Request Generator (PRG), according to NTCIP 1211 Scenario #2.

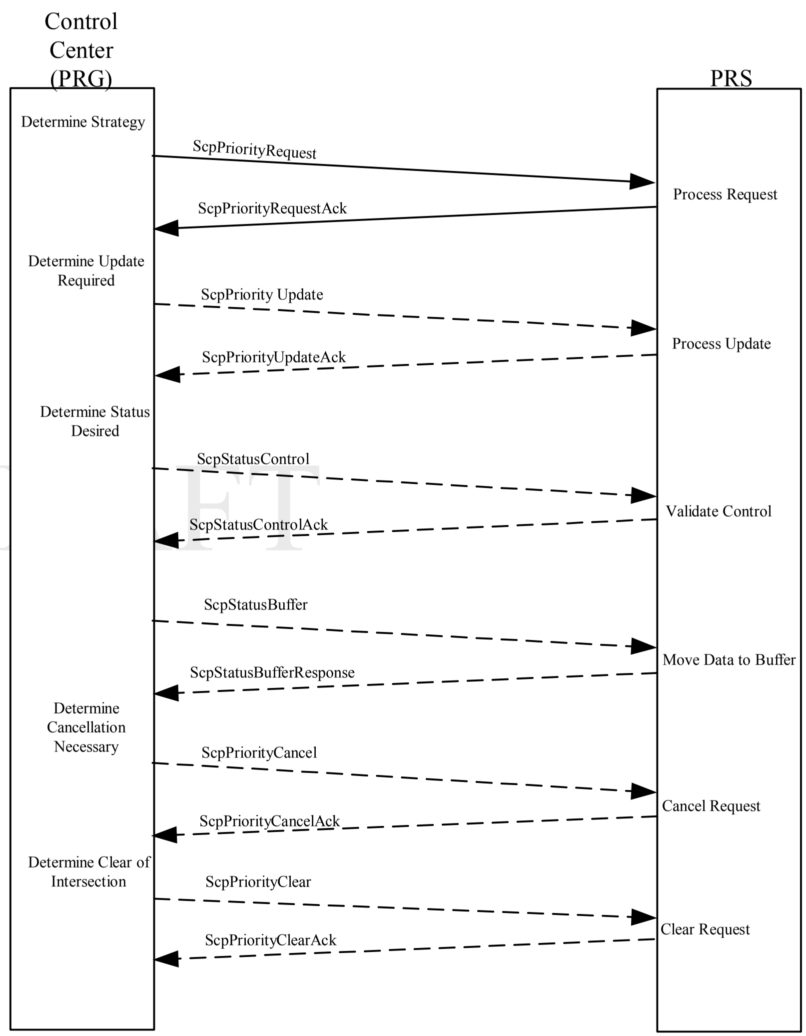
Assumptions:

1) The Control Center-based PRG has access to near-real-time vehicle information, such as vehicle location, schedule etc.

Narrative:

1) The PRG determines that the vehicle is approaching an SCP-equipped Scenario #2 intersection, and that the criteria are met to initiate a priority request. The PRG sends a ScpPriorityRequest message to the Priority Request Server (PRS).
    a. The PRS validates the request and sends a ScpPriorityRequestAck to the PRG.
    b. If the acknowledgement contains an error indication, the dialog ends, otherwise the PRG may optionally initiate any of items 2-5 below.
2) The PRG determines that it requires a status update on the priority request. The PRG sends a ScpStatusControl to the PRS.
    a. The PRS validates the status control and returns a ScpStatusControlAck message to the PRG.
    b. If the acknowledgement contains no error indication, the PRG sends a ScpStatusBuffer to the PRS.
    c. The PRS fills the status buffer and returns it to the PRG as a ScpStatusBufferResponse message.
3) The PRG determines that the priority request needs to be changed as a result of changed arrival time at the intersection and/or changes to the criteria matching. The PRG sends a ScpPriorityUpdate message to the PRS. The PRS processes the update and sends a ScpPriorityUpdateAck message to the PRG.
4) The PRG determines that the priority request needs to be canceled as the vehicle is not progressing towards the stop bar as planned, or has deviated from its route, or the vehicle is taken out of service, or experiences another locally defined state change requiring a cancellation. The PRG sends a ScpPriorityCancel to the PRS via the CC. The PRS processes the cancellation and sends a ScpPriorityCancelAck to the PRG, and the dialog ends.
    a. The CC performs local housekeeping actions and forward the acknowledgment to the PRG and the dialog ends.
5) The PRG determines that the vehicle is clear of the intersection, and the intersection requires explicit clears. The PRG sends a ScpPriorityClear

message to the PRS. The PRS clears the request and sends a ScpPriorityClearAck to the PIC and the dialog ends.



Normal Execution of "SCP Priority Request Scenario 2" Dialog

TCIP Dialog Definition Page 4

Dialog Name: SCP Priority Request Scenario 2

Business Area: SCP

Dialog Pattern: Signal Control & Prioritization

Message Name	Message Identifier	Role
Scp Priority Request	2011	Request a priority strategy from the PRS.
ScpPriorityRequestAck	2010	Acknowledge the priority request.
ScpPriorityUpdate	2009	Request that a previous priority request be modified.
ScpPriorityUpdateAck	2008	Acknowledge the priority update.
ScpStatusControl	2007	Request the PRS to prepare to provide status for a previous priority request.
ScpStatusControlAck	2006	Acknowledge the status control
ScpStatusBuffer	2004	Provide a buffer to obtain a priority request status
ScpStatusBufferResponse	2005	Return the status of the priority request to the PRG
ScpPriorityCancel	2002	Request the PRS to cancel a previous priority request
ScpPriorityCancelAck	2003	Acknowledge the priority cancellation
ScpPriorityClear	2012	Request the PRS to clear a completed priority request
ScpPriorityClearAck	2001	Acknowledge the priority clear.

Notes:

1) If the vehicle clears the intersection and the PRG is not configured to generate a priority clear to the PRS, the dialog ends from the PRG view point. The dialog ends from the PRS view point after a local timeout.

2) The criteria for determining when a priority update, status update, or priority cancel should be initiated are agency/vendor defined.

TCIP Dialog Definition Page 1

Dialog Name: SCP Priority Request Scenario 4

Business Area: SCP

Dialog Pattern: Signal Control & Prioritization

Purpose: This dialog defines the generation and processing of priority requests by an onboard vehicle Priority Request Generator (PRG), with direct communication to the Priority Request Server according to NTCIP 1211 Scenario #4.

Assumptions:

1) The PRG has already received SCP data via the "Upload SCP Data" dialog.
2) The PRG has access to data resident in the vehicle logic unit (VLU)such as vehicle location, speed, bearing, schedule, and passenger count (if so equipped).
3) Since the communications are direct between the PRS and PRG, none of the optional fields are used in the messages.

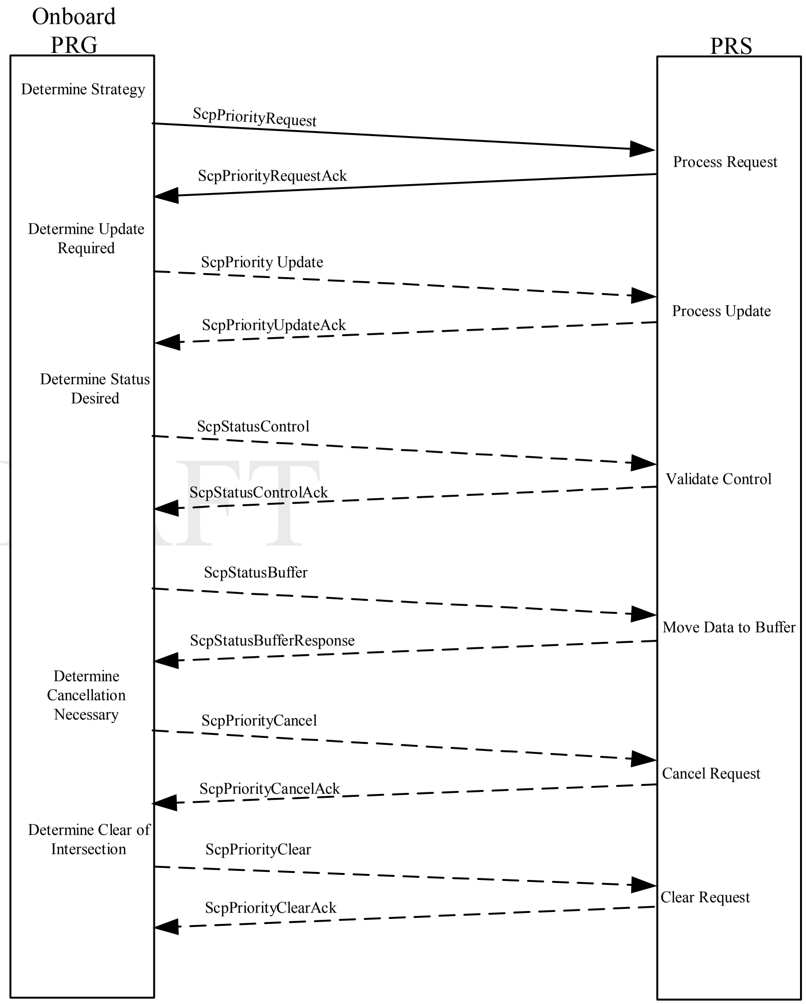
Narrative:

1) The PRG determines that the vehicle is approaching an SCP-equipped Scenario #4 intersection, and that the criteria are met to initiate a priority request. The PRG sends a ScpPriorityRequest message to the Priority Request Server (PRS).
    a. The PRS validates the request and sends a ScpPriorityRequestAck to the PRG.
    b. If the acknowledgement contains an error indication, the dialog ends, otherwise the PRG may optionally initiate any of items 2-5 below.
2) The PRG determines that it requires a status update on the priority request. The PRG sends a ScpStatusControl to the PRS.
    a. The PRS validates the status control and returns a ScpStatusControlAck message to the PRG.
    b. If the acknowledgement contains no error indication, the PRG sends a ScpStatusBuffer to the PRS.
    c. The PRS fills the status buffer and returns it to the PRG as a ScpStatusBufferResponse message.
3) The PRG determines that the priority request needs to be changed as a result of changed arrival time at the intersection and/or changes to the criteria matching. The PRG sends a ScpPriorityUpdate message to the PRS. The PRS processes the update and sends a ScpPriorityUpdateAck message to the PRG.
4) The PRG determines that the priority request needs to be canceled as the vehicle is not progressing towards the stop bar as planned, or has deviated from its route, or the operator manually initiates a cancel (if so equipped), or the vehicle is taken out of service, or experiences another locally defined state change requiring a cancellation. The PRG sends a ScpPriorityCancel to the PRS. The PRS processes the cancellation and sends a

ScpPriorityCancelAck to the PRG and the dialog ends.

5) The PRG determines that the vehicle is clear of the intersection, and the intersection requires explicit clears. The PRG sends a ScpPriorityClear message to the PRS. The PRS clears the request and sends a ScpPriorityClearAck to the PRG and the dialog ends.

Message Sequence Diagram Page 3



Normal Execution of "SCP Priority Request Scenario 4" Dialog

TCIP Dialog Definition Page 4

Dialog Name: SCP Priority Request Scenario 4

Business Area: SCP

Dialog Pattern: Signal Control & Prioritization

Message Name	Message Identifier	Role
Scp Priority Request	2011	Request a priority strategy from the PRS.
ScpPriorityRequestAck	2010	Acknowledge the priority request.
ScpPriorityUpdate	2009	Request that a previous priority request be modified.
ScpPriorityUpdateAck	2008	Acknowledge the priority update.
ScpStatusControl	2007	Request the PRS to prepare to provide status for a previous priority request.
ScpStatusControlAck	2006	Acknowledge the status control
ScpStatusBuffer	2004	Provide a buffer to obtain a priority request status
ScpStatusBufferResponse	2005	Return the status of the priority request to the PRG
ScpPriorityCancel	2002	Request the PRS to cancel a previous priority request
ScpPriorityCancelAck	2003	Acknowledge the priority cancellation
ScpPriorityClear	2012	Request the PRS to clear a completed priority request
ScpPriorityClearAck	2001	Acknowledge the priority clear.

Notes:

1) If the vehicle clears the intersection and the PRG is not configured to generate a priority clear to the PRS, the dialog ends from the vehicle PRG view point, and ends from the PRS view point after a local timeout.

2) The criteria for determining when a priority update, status update, or priority cancel should be initiated are agency/vendor defined.

TCIP Dialog Definition Page 1

Dialog Name: Download SCP Performance Data

Business Area: SCP

Dialog Pattern: Download

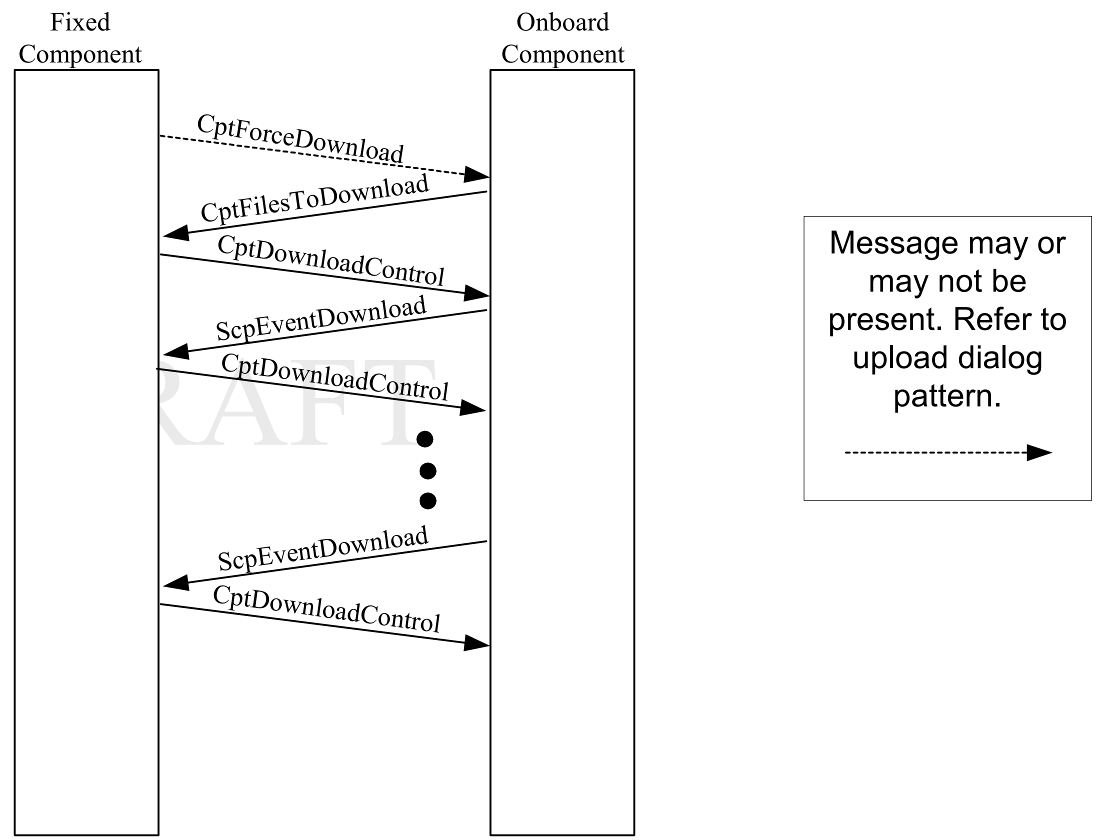
Purpose: Download data on the history signal priority events from the onboard component (VLU/MDT) to the fixed component which may be a CAD/AVL system or a data repository.

Assumptions:

Narrative:

1. The onboard component initiates the dialog based on a CptForceDownload message, or available files to download combined with WLAN availability, and sends a CptFilesToDownload message.
2. The fixed component determines what files are available, needed, or eligible for deletion and sends a CptDownloadControl message to the onboard component.
3. The onboard component deletes any files specified for deletion.
4. If there is no file specified to download the dialog ends. If the specified file is not available, the onboard component sends a CptDownloadRequestError message and the dialog ends. If there is a file specified and it is available, the onboard component sends the specified Scp Event Download (file) to the fixed component.
5. The fixed component receives and validates the Scp Event Download message and goes to step 2 above.

Message Sequence Diagram Page 2



Normal Execution of the "Download SCP Performance Data" Dialog.

TCIP Dialog Definition Page 3

Dialog Name: Download SCP Performance Data

Business Area: SCP

Dialog Pattern: Download

Message Name	Message Identifier	Role
CptForceDownload	Cpt 2017	Trigger onboard component to initiate a download process. Primary use is to download via a laptop instead of a wireless LAN.
CptDownloadControl	Cpt 2014	Used by the fixed component to control the download process.
CptDownloadRequestError	Cpt 2015	Used by the onboard component to notify the fixed component of a file request error.
ScpEventDownload	Scp	Conveys Scp History data from the onboard component to the fixed component.
CptFilesToDownload	Cpt 2013	Identifies files stored in an onboard component that are ready for download to the corresponding fixed component.

Notes:

TCIP Dialog Definition Page 1

Dialog Name: Subscribe SCP History Data

Business Area: SCP

Dialog Pattern: Subscription

Purpose: Provide a mechanism for a transit control center or a transit data repository (client) to obtain a historical record of signal priority events from a suitably equipped Traffic Management Center.

Assumptions:

The subscription may be a query or periodic type.

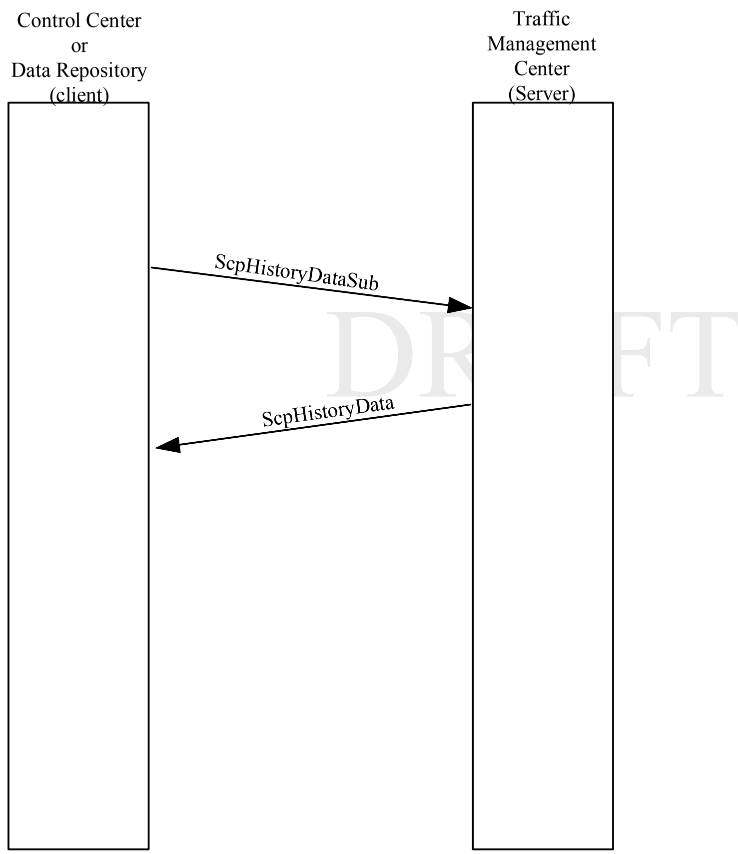
The client may be the control center or a data repository.

The server is an entity in the Traffic Management Center.

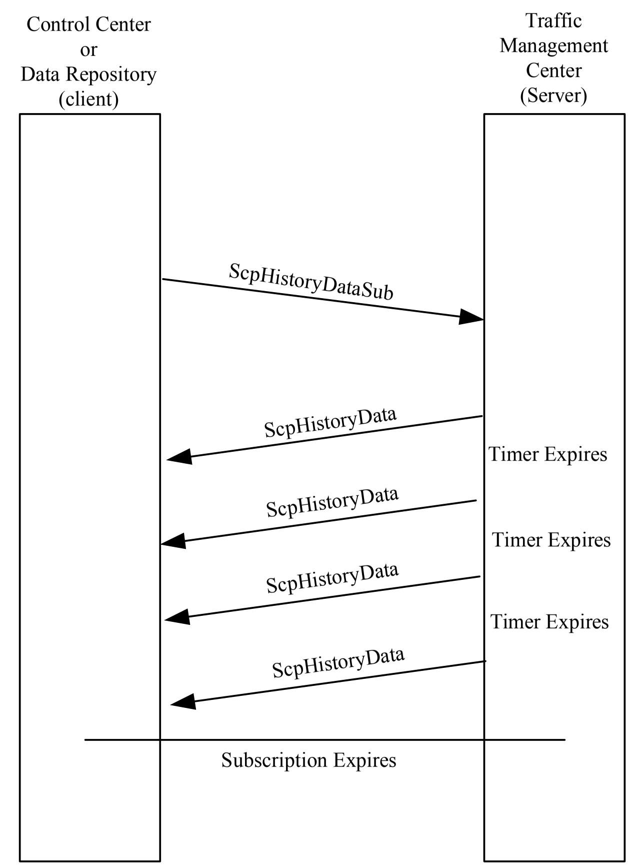
Narrative:

1) The client determines the intersections of interest, and forwards a ScpHistoryDataSub message (subscription request) . If the subscription is an ongoing periodic subscription, the reporting interval and subscription duration are also specified in the subscription request.
2) The TMC validates the subscription request.
    a. If the subscription request is invalid, the server sends a CptSubErrorNotice message to the client and the dialog ends.
    b. If the subscription request is a valid query, the server sends a ScpHistoryData message to the client and the dialog ends.
    b. If the subscription request is a valid periodic subscription request, the server sends a ScpHistoryData message to the client, containing history data currently on hand, and sends new ScpHistoryData messages at the interval specified in the subscription request.
3) The periodic subscription ends when:
    a. The subscription expires.
    b. The client sends a new HistoryDataSub message requesting a cancellation.
    c. The server sends a CptSubErrorNotice to the client.

Message Sequence Diagram Page 2



Normal Execution of the Query "Subscribe SCP History Data" Query Subscription.



Normal Execution of the "Subscribe SCP History Data" Periodic Subscription .

TCIP Dialog Definition Page 3

Dialog Name: Subscribe SCP History Data

Business Area: SCP

Dialog Pattern: Subscription

Message Name	Message Identifier	Role
ScpHistoryDataSub	SCP 2015	Used by the client to request signal priority history data from the server.
ScpHistoryData	SCP 2014	Used by the server to provide signal priority history data to the client.
CptSubErrorNotice	CPT 2000	Used by the server to notify the client that the subscription is terminated with an error status.

Notes:

TCIP Dialog Definition Page 1

Dialog Name: Subscribe Fleet SCP Information

Business Area: SCP

Dialog Pattern: Subscription

Purpose: Provide information on transit vehicle approaches to signal priority-equipped intersections to the Traffic Management Center. This subscription is required for NTCIP 1211 Scenario #3 generation of priority requests by the Traffic Management Center.

Assumptions:

1. The subscription may be or the event-driven or periodic type. Event-driven will result in more frequent, smaller messages to the TMC, while a periodic subscription would allow in formation to be buffered for several seconds and “batched” in a larger message.
2. The control center knows the locations of intersections, and stop bars as part of its configuration.

Narrative:

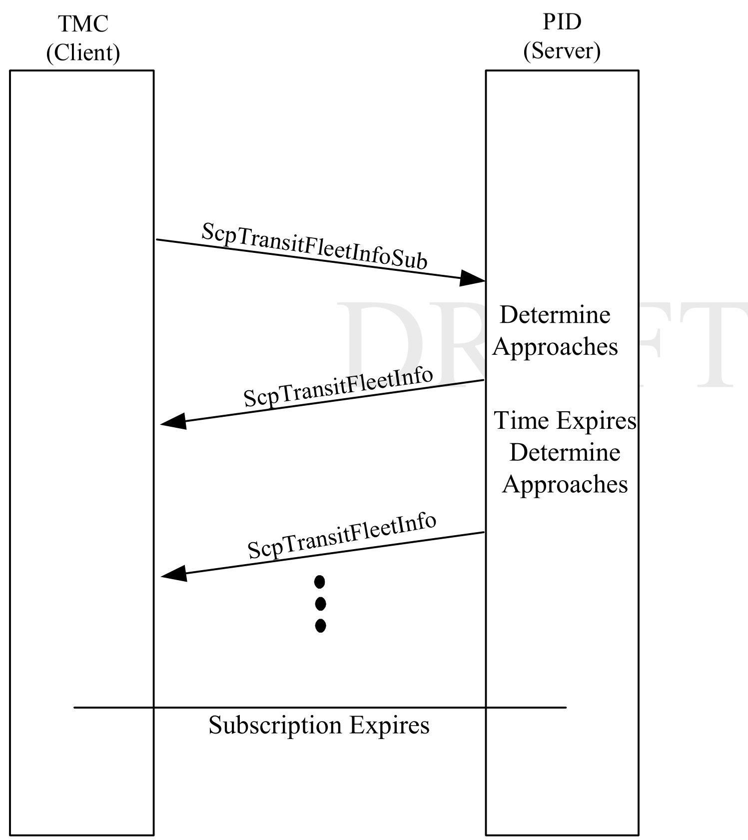
1. The client TMC sends a ScpTransitFleetInfoSub message.
2. The PID responds with a ScpTransitFleetInfo message containing current approach information.

a. If the subscription type is event, the Control Center waits for an approach to an SCP-equipped intersection, or abandonment of an approach to be detected and then reports it with a ScpTransitFleetInfo message.

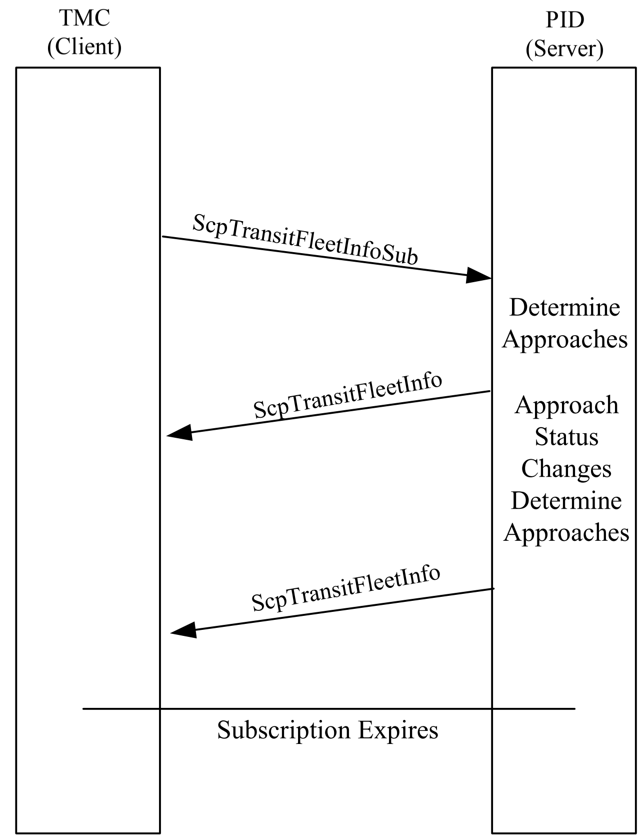
b. If the subscription type is periodic, the PID waits for the time to expire and then sends a ScpTransitFleetInfo message.

3. The dialog ends when the client sends a new ScpTransitFleetInfoSub message indicating “cancel”, or the subscription expires.

Message Sequence Diagram Page 2



Normal Execution of the "Subscribe Fleet SCP Information" Periodic Subscription



Normal Execution of the "Subscribe Fleet SCP Information" Event Subscription.

TCIP Dialog Definition Page 3

Dialog Name: Subscribe Fleet SCP Information

Business Area: SCP

Dialog Pattern: Subscription

Message Name	Message Identifier	Role
ScpTransitFleetInfoSub	Scp 2016	Used by the TMC to establish the subscription and request the approach information for SCP-equipped intersections.
ScpTransitFleetInfo	Scp 2017	Provides the information on transit fleet vehicle approaches to SCP-equipped intersections.

Notes:

ANNEX H

Base Types

This Annex will be added in a later version.