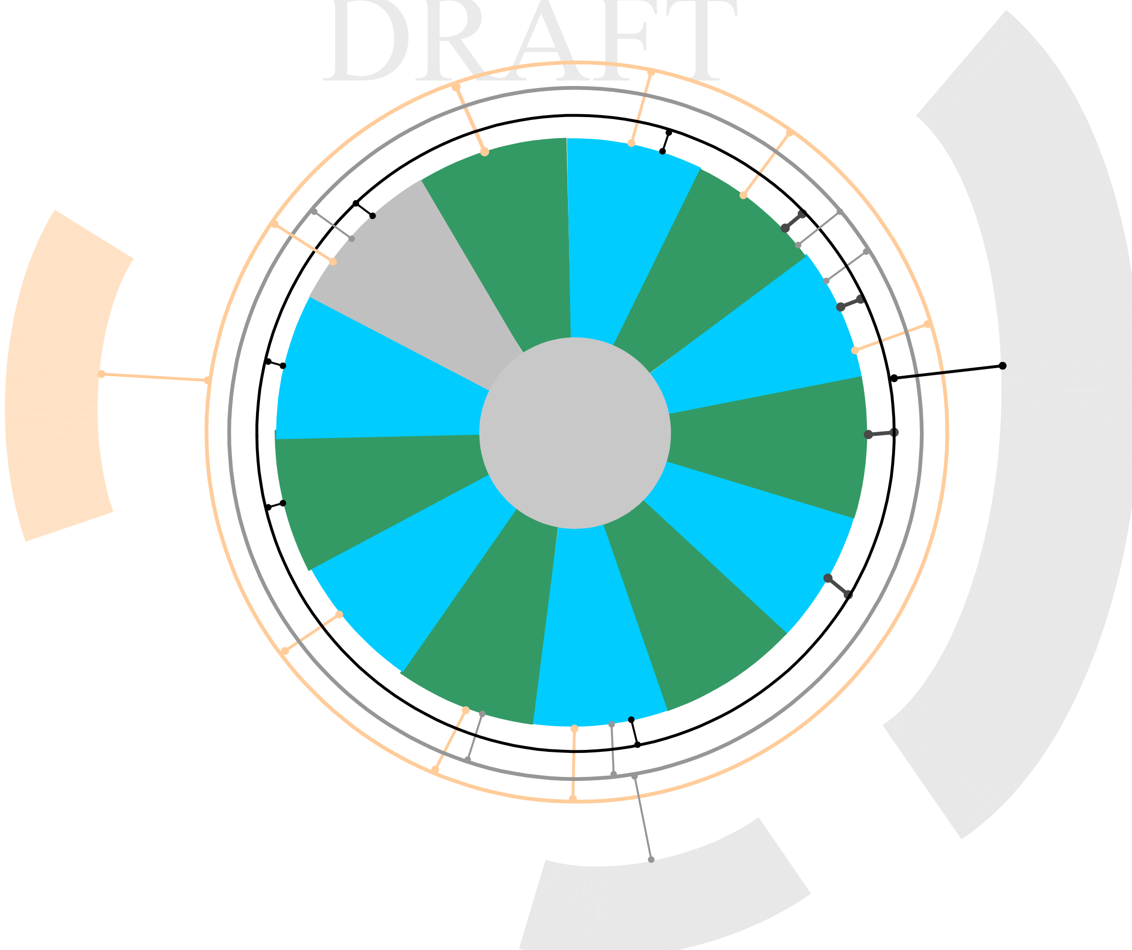
TCIP V2.5

Draft Transit Communications Interface Profiles

Version 2.5



This document was developed by APTA and two principal subcontractors ARINC Inc, and Critical Link, LLC under contract to the Federal Transit Administration.

This document contains material originally published in the NTCIP 1400 series standards, some of which has been modified or adapted for TCIP Version 2.5. Version 2.5 contains substantial changes from version 2.4, many of which are the result of comments from the APTA TCIP effort is overseen by the Task Force (Co-Chaired by Isaac Takyi and Jerome Lutin), and the APTA TCIP Technical Working Groups.

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. These standards were published by AASHTO, NEMA, and the ITE as: NTCIP 1400-1408. These standards have since been deprecated.

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 http://www.arincxchange.com/exchange/login.cfm (user name: apta guest, password: apta guest) .

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 file-based transfers of transit information using TCIP messages in situations where agencies do not want or need the automated information transfers specified in the dialogs.

File transfers allow standardized information to be transferred from one business system to another with or without network connection. File 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. TCIP 2.5 extends the file transfer paradigm to extend to data stores containing files which may be passed and read by an authorized application using XPATH/XQUERY.

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 from one system to another. Implementations that begin using TCIP for file transfers should account for the possibility of migrating to real-time transfers 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

Term	Definition
AASHTO	American Association of State Highway and Transportation officials. A professional organization for transportation professionals. Develops and promotes transportation standards.
ABS	See Authorized Business System
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.
ADUS	Archived Data User Service
Allocation Update	A data structure sent across the radio channel from the polling controller to the PTV announcing recent allocations and deallocations.
ANSI	American National Standards Institute
APTA	American Public Transportation Association. A non-profit association of transit agencies and suppliers of transit-related products and services.
APTS	American Public Transit Association
ASCII	American Standard Code for Information Interchange. A seven-bit binary code representation of letters, numbers and special characters. It is universally supported in computer data transfer.
ASN.1	Abstract Syntax Notation Revision One. 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	1. 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. 2. A message set for ATIS promulgated as SAE J2354.
Attribute	A quality or characteristic inherent in, or ascribed to someone or something.
Authorized Business System	A generic business system used to represent agency designated entities which may participate in a dialog.
AVL	Automatic Vehicle Location

Term	Definition
Bandwidth	the range of frequencies that can be used for transmitting information on a channel, equal to the difference in Hertz (Hz) between the highest and lowed frequencies available on that channel. Indicates the transmission-carrying capacity of a channel.
BER	Basic Encoding Rules or Bit Error Rate
Bit	Binary digit, a single basic computer signal consisting of a value of 0 or 1, off or on.
Bit Error Rate (BER)	The number of bits transmitted incorrectly. In digital applications, it is the ratio of bits received in error to bits sent.
BIT STRING	A series of binary (1,0) values.
Block	A vehicle work assignment.
Blocking	The process or organizing scheduled PTV trips into vehicle work assignments.
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.
BPS	Bits per second, transmission rate (speed) of data.
Byte	A group of bits acted upon as a group, which may have a readable ASCII value as a letter or number or some other coded meaning to he computer. It is commonly used to refer to eight-bit groups.
CAD	See CAD/AVL System
CAD/AVL System	A business system that dispatches, and monitors the activities of PTVs, and provides emergency/incident management capabilities.
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. A TCIP business area. See Control Center.
CCTV	Closed-circuit Television
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.

Term	Definition
Checksum	An arithmetic sum used to verify data integrity.
Codec	Coder/Decoder an entity that translates between two different message encoding formats. (e.g. TCIP XML encoded messages and TCIP Narrowband Encoded messages).
Common Public Transportation	A TICP Business Area. 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.
Control Center	A TCIP Business Area. 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.
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.
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.
Coordinated Universal Time	Time scale maintained by the Bureau Internationale de l'Huere (International Time Bureau) that forms the basis of a coordinated dissemination of standard frequencies and time signals.
CPT	Common Public Transportation.
CSS	See Customer Service System
Customer Service System	A business system used to support a transit agency's customer service department including managing mailings supporting the call center 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, etc. as attributes.
Data 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.
Device	See Controlled Device

Term	Definition
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.
Dispatcher	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.
DR	See Data Repository
DSRC	Dedicated Short-Range Communications
EIA	Electronics Industry Alliance
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.
FAR	See Fare System
Fare Collection	A TCIP business area. This business area involves the collection and processing of revenue of revenue from customers including the exchange of fare information.
Fare System	A business system that manages the fare and revenue collection functions.
FC	Fare Collection. A TCIP business area. See Fare Collection.
FCC	Federal Communications Commission
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.
FTP	File Transfer Protocol
Garage Operations System	A business system that supports the operations of a transit garage including marshalling operators and PTVs in and out of the garage to perform work.

Term	Definition
GIF	Graphics Interchange Format. A file format for exchanging graphical images.
GIS	Geographical Information System. A computer software application business system that organizes and processes information based on geographical coordinates as well as other attributes.
GOS	See Garage Operations System
GPS	Global Positioning System
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.
IEEE	Institute of Electrical and Electronics Engineers
IEEE 1512	IEEE Standard Sets: Communication Framework for Transportation Emergencies. Most TCIP references are to IEEE 1512. “Traffic Incident Management Message sets for Use by Emergency Management Centers”.
IETF	Internet Engineering Task Force
IM	Incident Management. A TCIP business area.
Implementation	A computer system, component, application business system, etc. that includes a TCIP interface.
Incident	Any event, crime, disturbance, equipment failure, weather anomaly, police investigation, traffic accident that disrupts or has the potential to disrupt transit service.
Incident Management	A TCIP business area. 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.
IP	Internet Protocol
ISO	International Organization for Standardization
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.

Term	Definition
ITU	International Telecommunications Union
Join Request	A Data structure sent across the radio channel from the PTV to the polling controller requesting that the PTV be allocated a polling slot number. Also used in response to a priority poll to notify the polling controller of an urgent message queued on the PTV.
LAN	Local Area Network
Leave Request	A data structure sent across the radio channel from the PTV to the polling controller requesting deallocation of its polling slot.
Load	Provide a large data message or messages to a vehicle or field device.
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. See SAE J2266.
Maintenance Management System	A business system that supports the maintenance of PTVs. Includes scheduling of maintenance and tracking of PYV health status.
Message	A grouping of data elements and/or data frames intended to be transmitted as a complete package of information in one direction.
Message Wrapper	A data structure sent across the radio channel in either direction conveying a TCIP –Narrowband –encoded message.
MM	See Maintenance Management System
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.
Namespace	According to W3C " An XML namespace is a collection of names identified by a URI reference which are used in XML documents as element types and attribute names".
NEMA	National Electrical Manufacturers Association.

Term	Definition
NTCIP	National Transportation Communications for ITS Protocol. A group of standards intended to promote interoperability among ITS components, and subsystems.
NTCIP 1211	An NTCIP Standard entitled “Objects for Signal Control Priority”.
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. A TCIP business area. See Onboard
Octet	A group of eight binary bits. Octets are a standard grouping for bits to be transmitted across a communications network.
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.
Onboard	A TCIP business area related to interactions among onboard PTV components.
PAR	See Parking Management
Parking Management	A business system that supports the operations of transit owned and/or operated parking garages.
Passenger Information	A TCIP business area. 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.
Pattern	A path consisting of a series of pattern segments followed by a PTV in executing a scheduled trip. A pattern segment is defined by a sequence of timepoints and stoppoints. A pattern segment may contain geographical trace points to facilitate mapping.
Pattern Segment	A portion of a path followed by a PTV in executing a scheduled trip. A pattern segment is defined by a series of timepoints and /or stoppoints.
PER	See Personnel Management System
Personnel Management System	A business system that manages transit employee information including work assignments.
PI	Passenger Information. A TCIP business area. See Passenger Information.
PICS	Profile Implementation Conformance Statement

Term	Definition
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.
Poll	A data structure sent across the radio channel from the polling controller to the PTV giving the PTV permission to transmit.
Poll Response	A data structure, containing PTV operating information, sent across the radio channel from the PTV to the polling controller in response to a poll.
Polling Controller	A fixed software entity that controls access to a data channel, by allocating and deallocating polling slot numbers to mobile units, and by issuing polls that allow PTVs to access the channel.
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.
Priority Poll	A data structure sent across the radio channel from the polling controller to the PTV to allow any PTV to notify the polling controller of an urgent message waiting.
Priority Request	A request to have traffic signal timing temporarily altered at a specified intersection to allow a PTV to operate more efficiently.
Priority Request Generator (PRG)	A logical entity that generates priority requests on behalf of one or more PTV(s).
Priority Request Server (PRS)	A logical entity in the traffic management environments (may be roadside or office based) that determines the disposition of priority requests.
PTSF	Public Transit Stop/Station Facility.
PTSFANN	A stoppoint/station based logical entity responsible for managing traveler information dissemination.
PTSFCFC	A stoppoint/station based logical entity responsible for managing fare collection activities.
PTSFSEC	A stoppoint/station based logical entity responsible for managing security functions.
PTV	Public Transit Vehicle. Any vehicle used to provide public transit service.

Term	Definition
PTVADH	A logical entity onboard a PTV responsible for monitoring and reporting schedule and route adherence.
PTVANN	A logical entity onboard a PTV responsible for managing passenger information dissemination.
PTVCFC	A logical entity onboard a PTV responsible for managing the fare collection activities.
PTVCOM	A logical entity onboard a PTV responsible for communications.
PTVDAT	A logical entity onboard a PTV responsible for PTV data.
PTVHEL	A logical entity onboard a PTV responsible for monitoring and reporting PTV health.
PTVLDS	An onboard PTV sensor that provides location to PTVLOC (e.g. GPS).
PTVLOC	A logical entity onboard a PTV responsible for monitoring and reporting PTV location.
PTVOPR	A logical entity onboard a PTV responsible for providing the operator interface.
PTVPAS	A logical entity onboard a PTV responsible for monitoring and reporting passenger boarding/alighting.
PTVSEC	A logical entity onboard a PTV responsible for managing security functions.
PTVTSP	A logical entity onboard a PTV responsible for managing transit signal priority functions.
Publisher	The provider of information in a subscription dialog.
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.
Real-Time Information Transfer	A conversation between two logical entities carried out using TCIP XML-encoded or Narrowband-encoded messages according to a TCIP dialog.
Roster	A grouping of operator work assignments into a weekly package.
Route	A publicly known PTV path over which service is provided- usually on a recurring basis. The pattern of stoppoints and timepoints serviced by PTVs operating on a route may vary by time of day, day of week, etc.

Term	Definition
Run	An operator work assignment.
Runcutting	The process of organizing scheduled PTV trips into operator work assignments
Running Time	The actual, expected, or scheduled time for a public transit vehicle to travel between two specified geographical points.
SAE	Society of Automotive Engineers
SAE J 1939	SAE family of standards entitled "Recommended Practice for Truck and Bus Control and Communications Network".
SAE J1587	SAE Standard entitled "Electronic Data Interchange Between Microcomputer Systems in Heavy Duty Vehicle Applications".
SAE J1708	SAE Standard entitled "Serial Data Communications Between Microcomputer Systems in Heavy-Duty Vehicle Applications.
SAE J2266	SAE Standard entitled "Location Referencing Message Standard".
SAE J2354	SAE Standard entitled "Message Sets for Advanced Traveler Information Systems (ATIS)".
SAE J2630	SAE Standard entitled "Converting ATIS Message Standards from ASN.1 to XML".
SCH	1. 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. 2. See Scheduling/Blocking/Runcutting System.
SCH	1, Scheduling a TCIP business area 2. A business system the model architecture that performs Scheduling/Blocking/Runcutting and runcutting functions.
Schedule	A collection of information that describes the planned movements of public transit vehicles.
Scheduling	A TCIP business area primarily involved with the creation and distribution of transit schedules and closely related information such as operator and vehicle work assignments.
Scheduling/Blocking/Runcutting System	A business system that creates transit schedules, and organizes planned trips within that schedule into operator and vehicle assignments.
SCP	Signal Control Prioritization. A TCIP Business Area that includes messages defined by NTCIP 1211.
SDO	Standards Development Organization

Term	Definition
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).
Service	The provision of transportation to transit customers.
Session Poll	A data structure sent from the polling controller to the PTV to allow a PTV without a polling slot to send a join request.
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.
SNMP	Simple Network Management Protocol
SP	Spatial Positioning. One of 8 business areas defined by TCIP.
Spatial Positioning	A TCIP Business Area. 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.
STMP	Simple Transportation Management Protocol
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.
Subscriber	The consumer of information in a subscription dialog.
Subscription	A relationship between an information consumer (subscriber) and an information provider (publisher) 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.
TCIP	Transit Communications Interface Profile.
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol

Term	Definition
TEI	Transit Employee Interface – a device that allows a transit employee to interact with agency data networks. An example is a handheld wireless device used by a supervisor to enter an incident report.
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*")
Timepoint	A point along a public transit vehicle's route where the vehicles scheduled time to transit that point is identified in the schedule.
TMDD	Traffic Management Data Dictionary. A standard promulgated by ITE/AASHTO as standard number TMI.03.
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. 3. Move data between business systems and/or logical entities.
Transit Signal Priority	A TCIP business area related to obtaining preferential treatment for PTVs at signalized intersections.
Traveler Information System	A business system that provides information directly to transit travelers via a variety of communications media.
Trigger	1. The initiation of a series of actions (as of a dialog). 2. The event that initiates a series of actions.
Trip	1. 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. 2. A single scheduled instance of a PTV providing service on a route in a designated direction. 3. A journey planned or undertaken by a transit customer, or a portion of such a journey.
TRV	See Traveler Information System
TSP	Transit Signal Priority. A TCIP business area. See Transit Signal Priority.
UDP	User Datagram Protocol
UDP/IP	User Datagram Protocol/Internet Protocol

Term	Definition
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.
Unload	Remove a large data message or messages from a vehicle or field device to a fixed business system.
URI	Universal Resource Identifier
UTC	Universal Time Coordinated – See Coordinated Universal Time
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.
VIN	Vehicle Identification Number
WAN	Wide Area Network
WLAN	Wireless Local Area Network
XML	Extensible 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

This section defines the minimum requirements to be met by an interface to claim TCIP conformance. TCIP is an interface standard and not a functional specification for agency business systems, components, or logical entities. Consequently, the fact that an interface is TCIP conformant will not in and of itself, ensure that agency functional requirements are met.

TCIP does not require support for all dialogs or messages to achieve conformance. Instead TCIP defines classes of conformance that allow a transit agency to pick and choose the class(es) of conformance for each interface that best meet agency needs. Thus an agency can select a group of dialogs to be supported by a class 1 interface, or a group of messages to be provided as files or documents via a class 2 interface, and/or can use a class 3 interface to implement polled narrowband communications between the PTV and the control center.

3.1 Basic Conformance Requirements

A conformant TCIP Interface consists of 2 elements:

- A Profile Implementation Conformance statement (PICS) which defines what parts of TCIP the interface supports.
- An Implementation which provides the actual physical and logical interface to other system(s).

A business system, or component may provide more than one interface. The conformance of each such interface is determined independently, requiring a separate PICS and being treated as a separate implementation.

3.1.1 Basic PICS Requirements

A TCIP PICS Shall:

- Specify the TCIP Version or Versions supported by the Implementation.
- Specify the TCIP Class or Classes of conformance supported by the implementation.
- Comply with PICS requirements associated with each specified conformance class.
- Specify what optional fields are supported for each TCIP message generated or sent by the implementation.
- Specify what optional fields are necessary for the implementation to function for each TCIP message received by the implementation. Commentary: Although a field may be specified as optional in TCIP, the information in that field may be critical to the function of a specific implementation. For example an implementation may require that patterns in a schedule be named, as well as numbered, whereas TCIP requires numbers and specifies that names are optional.
- Specify any supported locally defined extensions to TCI messages send or received by the implementation.
- Specify the IP addresses and port numbers or alternative network and transport addressing parameters used by the implementation, both to identify itself and to identify other interfaces with which it communicates. Commentary: The implementation may be delivered with an initial set of addresses and a configuration capability, in which case the configuration procedure should also be defined here.
- Identify any respect in which the interface does not meet TCIP requirements. Note: Generally anything listed here is considered a nonconformant feature. Failure to list a nonconformant renders the interface nonconformant in its entirety. Commentary: the intent is to ensure that nonconformances are clearly identified by the developers so that agencies can determine whether nonconformant features are acceptable in their environment. Since failure to identify a nonconformant feature in the PICS renders the entire interface nonconformant, a supplier may not claim that the resulting interface is in any respect a TCIP interface.

3.1.2 Basic Implementation Requirements

A TCIP Implementation of any class shall:

- Comply in every respect with its PICS.
- Comply with all requirements associated with each conformance class for which conformance is claimed in the PICS.

3.2 TCIP Conformance Classes

TCIP Interfaces shall support one or more conformance classes. The TCIP Conformance Classes are:

- Class 1A Dialog Based Interface with XML Encoded Messages.
- Class 1B Dialog Based Interface with TCIP Narrowband Encoded Messages.
- Class 2A File/Document Based Interface with XML Encoded Messages
- Class 2B XQUERY/XPATH Based Interface with XML Encoded Messages.
- Class 3A TCIP Polling Protocol Interface – Polling Controller Side
- Class 3B TCIP Polling Protocol Interface – PTV Side

Conformance requirements for each conformance class are defined in subsequent sections.

3.2.1 Dialog Based Conformance Classes (Class 1)

All Class 1 PICS shall:

- Define what dialogs are supported by the implementation
- Define what entities within each dialog are supported by the implementation. Commentary: Some implementations may support both sides or the same dialog. For example a Data Repository may subscribe to some information from its source and then publish the same information to other applications.

All class 1 implementations shall conform in every respect with the dialog definitions in Annex D and the Pattern definitions in Section 7 for the dialogs and patterns included with the implementation.

3.2.1.1 Dialog based Interface with XML Encoded Messages (Class 1A)

All Class 1A Implementations shall:

- Provide all required fields in the XML Schema for every transmitted message.
- Comply with any "WITH COMPONENTS" fields present in the ASN.1 definition of any transmitted message.
- Accept all required and/or optional fields in any XML message instance of any message defined to be received by the Implementation based on the dialogs/entities supplied by the Implementation as specified in the PICS.
- Accept maximum length message instances for all message defined to be received by the Implementation as specified in the PICS.
- Provide internal recovery mechanisms from any error state generated by the dialog (e.g. lost/bad messages, no response etc.).
- For any message specified to be received by the Implementation, accept any message instance that complies with TCIP.XSD.
- For any message instance transmitted by the Implementation, include the minimum set of XML message instance attributes as specified in section 3.2.4.
- For any XML message instance transmitted by the Implementation, the message shall be well-formed and valid.

3.2.1.2 Dialog Based Interface with Narrowband Encoded Messages (Class 1B)

All Class 1B PICS shall:

- Define any local limitations on sequence lengths for supported messages.
- Commentary: local implementation may shorten the maximum number of items allowed in a list across a narrowband interface. For example:

  ExampleMsg::=SEQUENCE{

  a-infosets : SEQUENCE (SIZE(1..100)) OF Item-A

  }

  may contain a provision in the PICS of the form:

  "Narrowband encoded ExampleMsg instances are limited to a maximum of 15 a-infosets per message across this interface."

  If the Implementation has no local limitations, the PICS shall state:

  "This Implementation supports message instances containing maximum length sequences for all narrowband-encoded messages supported by this interface.

- Define any local limitations on overall narrowband-encoded message length, which may be asymmetrical. For example the PICS may contain

  "Narrowband-encoded message instances transmitted by this Implementation are limited to 200 octets per message instance. Narrowband-encoded message instances received by this Implementation are limited to 300 octets per message instance."

  If the Implementation has no local limitation, the PICS shall state:

"This Implementation supports maximum-length message instances for all narrowband-encoded messages supported by this interface."

All Class 1B implementations shall:

- Provide all required fields in the ASN.1 definition for every transmitted message.
- Comply with any "WITH COMPONENTS" fields present in the ASN.1 definition of any transmitted message.
- Accept all required and/or optional fields in any narrowband message defined to be received by the Implementation based on the dialogs entities supplied by the Implementation as specified in the PICS.
- Provide internal recovery mechanisms from any error state generated by the dialog (e.g. lost/bad messages, no response etc.).

3.2.2 File/Document Transfer Interfaces (Class 2)

All Class 2 PICS shall:

- Specify what TCIP Messages are provided and/or consumed by the Implementation.
- Specify any operating system or operating system specific attributes of the files(s) or data stores providing or receiving the TCIP messages.
- For any provided TCIP message, define what optional fields will, will not or may be provided in the file or data store.

All Class 2 implementations Shall:

- For any consumed TCIP message accept any message instance that complies with the TCIP.XSD Schema.
- For any provided TCIP message, include the minimum set of XML message instance attributes as specified in section 3.2.4.

3.2.2.1 File/Document Based Interface with XML Encoded Messages (Class 2 A)

Class 2A Pks Requirements:

- If the Implementation provides TCIP messages as files, the PICS shall specify the procedure for producing the file on removable media or otherwise exporting and transferring the files to another system.
- If the Implementation consumes TCIP messages as files, the PICS shall specify the procedure importing the files from another system.

Class 2A Implementation Requirements:

- For any XML message instance transmitted by the Implementation, the message shall be well-formed and valid.

3.2.2.2 XQUERY/XPATH Based Interface with XML Encoded Messages (Class 2 B)

All Class 2B PICS shall:

- Specify what version(s) of XPATH/XQUERY are supported by the Implementation.
- Specify what communications networks and protocols the Implementation supports.
- If the Implementation initiates queries, specify the procedure for triggering/initiating a query.
- If the Implementation accepts queries, specify the procedure for defining user authorizations.

All Class 2B Implementations shall:

- Include a prolog with every XQUERY instance that include the SchemaImport and NamspaceDecl items specifying TCIP.XSD, TMDD.XSD, and LRMS.XSD. TCIP.XSD shall be the default namespace.
- Implementation providing a data store shall provide TCIP XML messages as documents to be queried.
- For any XML message instance transmitted by the Implementation, the message shall be well-formed and valid.

3.2.3 TCIP Polling Interface

A Class 3A TCIP Interface shall meet all requirements for the Polling Controller specified in Section 10.

A Class 3B TCIP Interface shall meet all requirements for the PTV/VLU specified in Section 10.

3.2.4 XML Message Instance Attributes

All TCIP XML Message/Document instances used in implementation classes 1A, 2A, and 2B shall include all of the attributes listed in Table 3.2.4 at the beginning of the message/document. Implementations that generate messages/documents may provide additional attributes. Agencies may require additional attributes to be included by any purchased implementation. Agencies which use non-IP networks may specify alternate content for the sourceip and source port fields.

Table 3.2.4
Required Attributes in TCIP XML Messages & Documents

Attribute Name	Type	Example
created	xs:datetime	created= “2004-01-10T:19:59:01”
schversion	xs:string	schversion = “TCIP 3.0”
sourceapp	xs:string	sourceapp= “KCMETRO DIST Db”
sourceip	xs:string	sourceip= “1.2.3.4.5.6.7.8.9.11.12.13.14.15.16”
source port	xs:integer	source port= “160”
No Name Space Schema Location	xs:string	No NamespaceSchemaLocatin=”tcip.xsd”

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.

4.1 Structure of TCIP

TCIP defines the messages, data frames, and data elements to be used for information transfer on either a file-oriented or dialog-oriented (message exchange). 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 4.1 depicts the building block approach to TCIP.

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 NAME30, 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, and 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 file transfer builds on the messages, but not the dialog patterns, because these interactions are defined by vendors and agencies.

The agency architecture builds on the agency's legacy systems- on TCIP-File Transfer (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.

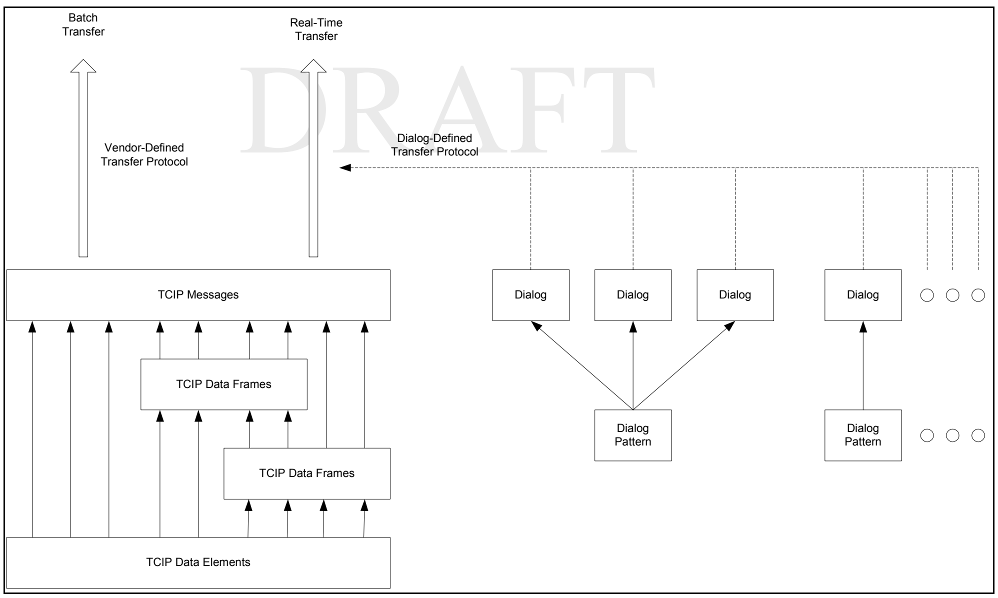


Figure 4.1
Depicts the RelationshipsAamong TCIP Components.

4.2 Transit Model Architecture

4.2.1 Overview

The TCIP model architecture, identifies key transit agency elements pertinent to TCIP and their interfaces. It also provides a useful structure for illustrating the building block approach to interface development provided by the TCIP dialog. ***The model architecture is not intended to represent the actual configuration of any transit agency, nor is it intended to recommend any specific approach to agency or regional architecture development. The sole purpose of the Model Architecture is to provide a conceptual basis for the discussion of agency business processes and interfaces.***

The components of the Model Architecture are:

- Logical functions (or groups of logical functions),
- Dialogs that interface the logical functions,
- Interfaces to infrastructure (e.g. vehicle, stoppoint, hardware),
- External systems, and
- Physical elements

The mapping of logical functions into physical elements (packaging) is intended only to supply a coherent arrangement of the logical functions – and is not intended as guidance for packaging in product development or in agency architecture.

The wedges in the circle in Figure 4.2.1 depict typical Transit Agency Business Systems. The circle at the core of the Model Architecture represents the Transit Data Network which enables communications among the business systems. The rings represent wired and wireless wide area networks that allow Business Systems to communicate with entities in the field, on board PTVs and external to the transit agencies, depicted as arcs in Figure 4.2.1.

Business Systems represent a group of related logical functions that are treated as a single entity within the Model Architecture. A Model Architecture Business System is thus a logical entity which may be hosted in one or more physical elements. For discussion purposes, a Business System is treated as both a single physical element and logical entity. The internal architecture of a business system is left to the developer and is outside the scope of the Model Architecture. Thus, the Model Architecture does not define logical entities, physical elements interfaces, or data flows within a Business System. Similarly, agencies may adopt agency, local, or regional architectures that reflect other organizations of logical functions into business systems, and still use applicable TCIP dialogs and file transfers on an a-la-carte basis to exchange information between those Business Systems.

These business systems communicate with each other either:

- Via file/document transfers which may be conducted through the transit agency data network, or
- Via physical file/document transfers using removable media, or
- Via dialogs which define message-based conversations between business systems.

Some agency architectures include a data repository – usually a large database, or group of databases. In such agencies, the repository serves as an intermediary in these conversations between business systems. This allows a business system to produce or consume information asynchronously from other business systems, eliminates the need for an information producing system to communicate with all of the information consumers, and provides a platform for fusing data from multiple sources. Data repositories generally require technically sophisticated staff to set up and administer, and consequently are not found in many smaller agencies.

Transit Agencies operate a variety of field systems and devices. These devices maybe monitored, controlled, or both. Field devices include elevators, escalators, door alarms, passenger information displays, vent fans, and a wide variety of other device types. TCIP provides a limited set of dialogs for communicating with logical entities that act as controllers or managers of these devices. The Model Architecture identifies these field device controller/manager entities, but does not specify their physical configuration. These logical entities may be packaged with the field device or may be packaged separately in a unit that controls/manages a group of devices (e.g. for a passenger station). Other agency-specified standards and/or proprietary protocols are used to interact with field devices in parallel with TCIP.

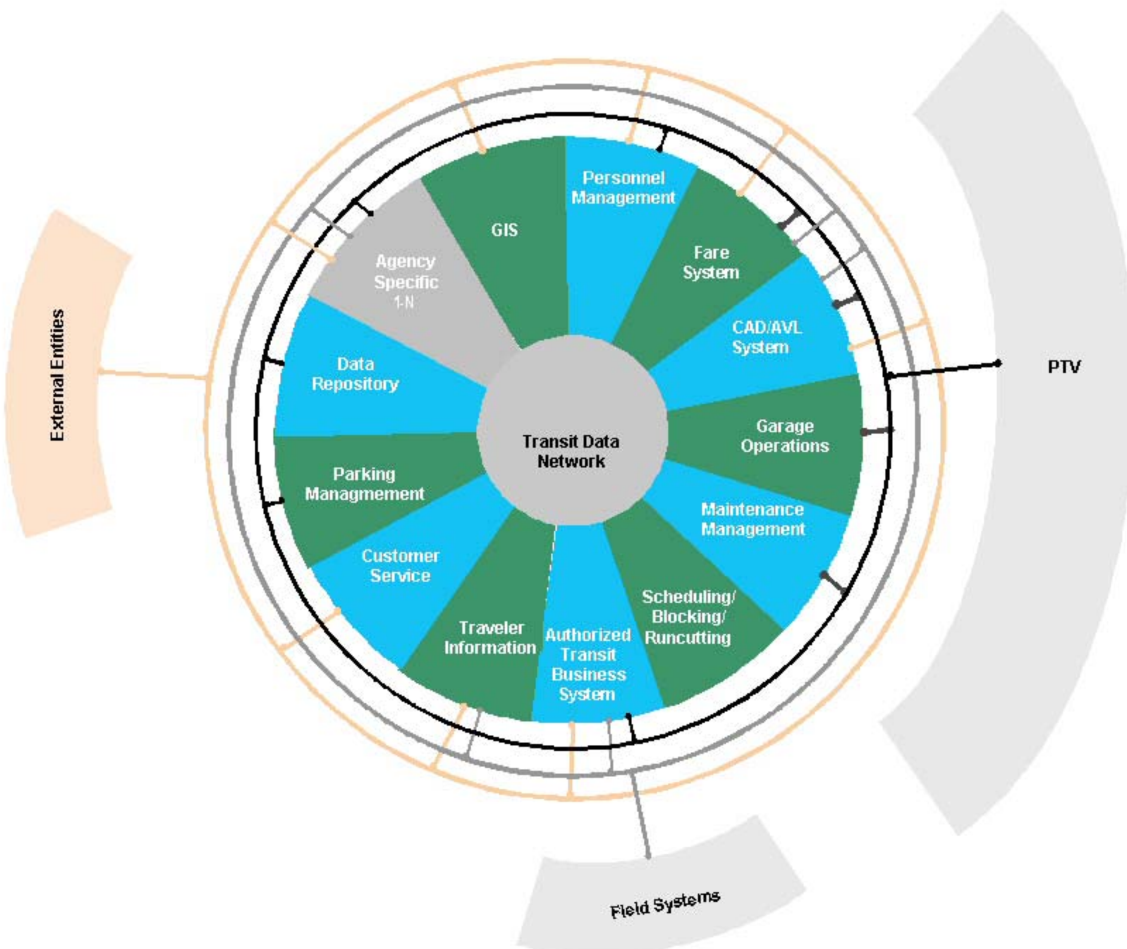
Public Transit Vehicles (PTVs) contain a variety of logical functions that are packaged very differently from one agency (or even one bus type within an agency) to another. The Model Architecture identifies these logical entities, and identifies some, but not all, possible physical configurations of these entities. TCIP provides dialogs for communication between PTV logical entities. TCIP provides dialogs for communications between PTV logical entities and fixed business systems, and dialogs for some communications between onboard logical entities which are likely to be separately packaged.

Transit Business Systems also communicate with entities outside of the transit agency itself. These include external Advanced Traveler Information Systems, Traffic Management Systems, Information Service Providers, other transit agencies, public safety agencies, traveler-owned Internet appliances, and other systems. These conversations are generally governed by standards promulgated by other industries, however TCIP dialogs do include support for Transit Signal Priority interactions with Traffic Management Systems consistent with IEEE 1512.

In parallel with the development of TCIP, an effort has been underway to develop a National ITS Architecture (NIA). Since these efforts were conducted independently, there is an imperfect match between TCIP and the NIA. Nonetheless, there is substantial interest in a mapping between NIA and TCIP. Such a rough mapping contained in Annex I "TCIP-NIA Mapping". This Annex maps TCIP logical entities to the closest match NIA 'Pspec' and TCIP

Dialogs to the closest match NIA data flows. Note that NIA data flows are unidirectional, and dialogs are bi- or multi-directional so dialogs generally map to more than one data flow.

Figure 4.2.1
Top Level Transit Model Architecture.



4.2.2 Business Systems

Transit Business Systems may be developed by the agencies themselves or by commercial providers. Generally commercial systems are tailored to some extent for each agency, and agency-developed systems tend to be highly customized. This implies that there may be significant differences in functionality and operation from one agency to another. Furthermore, different agencies and vendors package functionality into business systems with different names and groupings. Thus the model architecture and associated assignments of functions to business systems in this model is neither normative, nor definitive.

TCIP assumes that business systems maintain logs of events and transactions and provide archiving and retrieval capabilities which may be ADVS compliant. TCIP does not standardize the logging functions of business systems, or define their specific requirements.

Table 4.2.2 summarizes the business systems illustrated in figure 4.2.1, lists their major functions and typical agency architectural variations.

Table 4.2.2

Transit Agency Business Systems

Name	Typical Functions/Contents	Typical Variations
Agency Specific (1-N)	▪ Any Agency defined Business Systems	
Authorized Business System (ABS)	▪ May be agency designated to participate as a business system in most dialogs.	
CAD/AVL System (CAD)	▪ Track PTVs ▪ Provide Dispatcher Interface ▪ Provide PTV Adherence Monitoring Interface ▪ Provide Alarm Management ▪ Manage Exceptions to Operating Plan ▪ Manage Incidents ▪ Provide Operating Status Information to other Business Systems	
Customer Service (CSS)	▪ Complaint investigation o Completed trip history o Playback of PTV operations ▪ Itinerary creation ▪ Parking information ▪ Call taking ▪ Demand (paratransit) trip booking ▪ Customer eligibility (e.g. senior, paratransit)	▪ Customer Service System and Traveler Information System may be combined

Table 4.2.2

Transit Agency Business Systems

Name	Typical Functions/Contents	Typical Variations
Data Repository (DR)	▪ PTV Data Load/Unload ▪ Security Images Archiving ▪ Transit Facilitates Information Store ▪ Personnel Data Store ▪ Schedule Data Store ▪ Software & Configuration Data Store ▪ Historical Operating data Store	Data Repository may not be present, & data stores distributed among other Business Systems
Fare System	▪ Manage fare zones/fare tables, etc. ▪ Fare data and nightly reconciliation with PTVs/Fare vending machines ▪ Bank reconciliation ▪ Fare type/count/location Tracking & Reporting ▪ Bad card list ▪ Card issuance management	▪ Some or all functions may be performed by regional reconciliation system ▪ Some functions may be contracted out to a commercial services provider
Garage Operations (GOS)	▪ Bind available cars/vehicles to planned trips/runs/blocks ▪ Manage pull-outs/pull-ins ▪ Manage fitness for service checks (operators) ▪ Manage fitness for service checks (equipment) ▪ Manage last minute changes to operator and vehicle assignments	▪ May be combined with Maintenance Management
Geographical Information System (GIS)	▪ Transit Facilities Information ▪ Transit Route/Pattern Information ▪ Stoppoint inventory ▪ Fare zone geometry ▪ Ridership history by location ▪ Area maps ▪ Jurisdictional boundaries	GIS may be provided by local or regional government.
Maintenance Management System (MM)	▪ Track all past performed maintenance events ▪ Forecast and plan scheduled maintenance ▪ Forecast and plan vehicle car availability ▪ Track vehicle car work orders ▪ Track/forecast component subsystem performance/reliability ▪ Track/forecast spare parts requirements and requisitions	▪ May be combined with Garage Operations

Table 4.2.2
Transit Agency Business Systems

Name	Typical Functions/Contents	Typical Variations
Parking Management (PAR)	▪ Transit Agency Operated Parking Garage Management System	
Personnel Management (PER)	▪ Payroll calculations ▪ Manage operator "pick" of assignments ▪ Generate "Extra List" ▪ Bind operators to operator assignments based on pick	▪ Pick system may be integrated with garage operations
Scheduling/Blocking/Runcutting (SCH)	▪ Create Schedule o Define Patterns o Define Trips o Define Timepoints o Define Transfers ▪ Blocking/Runcutting o Define Operator Assignments (unbound) o Define Vehicle Assignments (unbound) ▪ Reporting (various output data filters and organizations)	Timepoints and/or stoppoints may be defined in scheduling system, GIS, data repository, or ad-hoc tool (e.g. spreadsheet).
Traveler Information (TRV)	▪ Itinerary creation ▪ Parking information ▪ Fare information (for planned trip) ▪ Next Bus information o Timetable o ETA ▪ Passenger Information Display (PID) monitoring & control	▪ Traveler Information System and Customer Service System may be combined

4.2.3 Public Transit Vehicles

Public Transit Vehicles exhibit differing physical architectures from one agency to another, and in many cases between vehicle types or brands within an agency. The PTV Model Architecture shown in Figure 4.2.3. The diagram shows a typical set of physical components. To highlight the variations, logical entities that are most likely to be packaged separately from the VLU are in green italics. Table 4.2.3 lists the PTV's logical entities and their purposes.

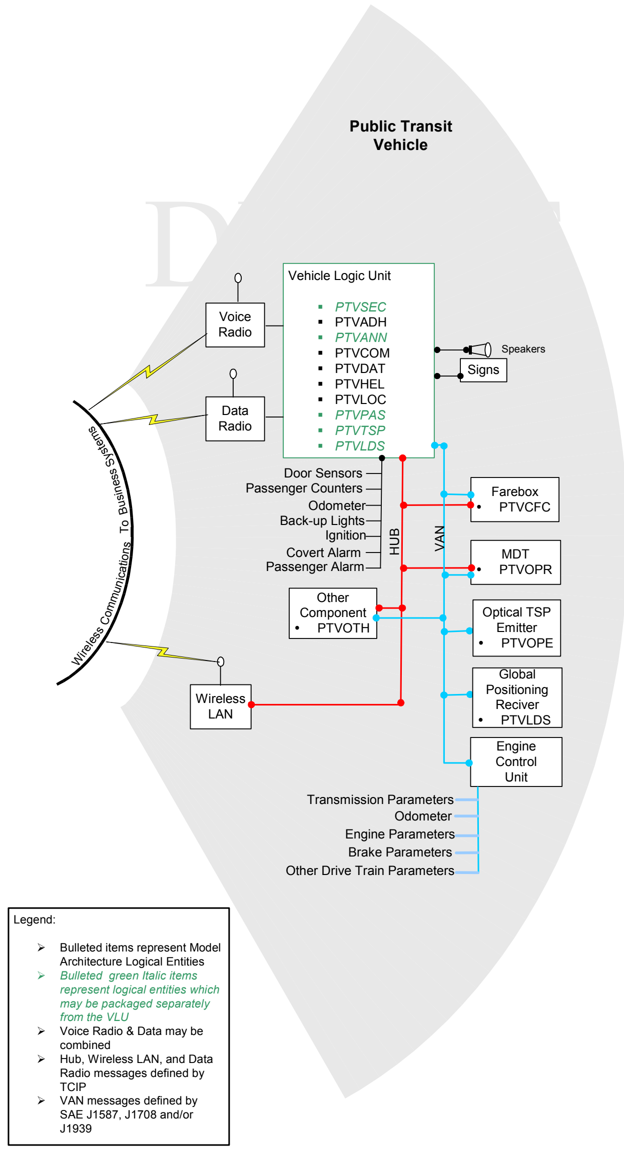


Figure 4.2.3
PTV Model Architecture

PTV legacy communications are usually based on the SAE J1708/J1587 standards. Newer vehicles are being fielded now using the higher speed J1939/J1587 Vehicle Area Network (VAN). In either case the network shown in blue on Figure 4.2.3 conveys SAE-standardized information. These standards are designed to allow arbitrary combination of compliant devices to co-exist on the VAN and to exchange information.

As information technology has progressed, more and more PTVs are being equipped to communicate via high-speed wireless local area networks (WLAN). WLAN communications typically uses the Internet Protocol (IP) at layer3 with the potential to implement a variety of higher-level protocols as is done with office local area networks. This high-speed connection to the fixed side data networks can be shared by a variety of onboard PTV components using a hub. This hub allows various components to access the WLAN, but also enables hub-connected components to communicate with each other using IP. This capability is represented by the red network in Figure 4.2.3.

PTVs also have data communications with fixed business systems when outside of the WLAN coverage area either through the transit agency's private radio network, or through public networks provided by the common carriers (phone companies). These facilities are generally more expensive and/or bandwidth limited, and agencies generally avoid sending large data transfers via these mechanisms.

TCIP dialogs specify the conversations to be carried out between:

- Onboard hub-connected components using XML (red network)
- Onboard components communicating to fixed business systems via the WLAN
- Onboard components communicating to fixed business systems via the private or public data network.

TCIP dialogs do not specify the communications occurring on the VAN (blue network), however, beginning with TCIP Version 2.5, the dialogs will note where related VAN communications occur within a dialog's execution.

Table 4.2.3 summarizes the major communicating components on PTV along with their major functions and typical variations.

Table 4.2.3

PTV Logical & Physical Entities

Name	**Typical Functions/Contents**	**Typical Variations**
Engine Control Unit	▪ Monitor drivetrain and other vehicle parameters and distribute their values via VAN	
Farebox (PTVCFC)	▪ Collect and record fares via various media ▪ Report fare information to fare system for end-of-day reconciliation ▪ Accept login from MDT or provide login to MDT ▪ Provide passenger counts to other onboard entities based on fares collected ▪ Accept passenger counts from other onboard entities to validate number of fares collected.	▪ Login exchanges are not available on all fareboxes ▪ Passenger count information is not provided to the VLU by many fareboxes

Table 4.2.3 PTV Logical & Physical Entities		
Name	**Typical Functions/Contents**	**Typical Variations**
Location Data Source (NIA Entity) (PTVLDS)	Onboard PTV sensor that provides location to vehicle (PTVLOC).	May be packaged with the VLU or separately. May be connected via VAN
Mobile Data Terminal (PTVOPR)	▪ Provide operator logon/logoff with notification to other entities ▪ Provide relief operator sign on with notification to other entities ▪ Provide two-way operator – dispatcher canned message and text message exchange ▪ Provide operator logon/logoff notification to other onboard components ▪ Provide input/output screen capability for other onboard devices	▪ May be combined with the Vehicle Logic Unit
Optical TSP Emitter	Onboard PTV Light emitter that conveys transit signal priority requests to suitably equipped intersections.	
PTV Communications Manager (PTVCOM)	This entity controls the PTV radio, provides updates on wireless LAN availability, and performs other communications management activities as specified by the local agency for the VLU.	
PTV Count Passengers (PTVPAS)	Onboard PTV passenger counting entity.	May be packages with the VLU, Farebox or separately
PTV Manage Security (PTVSEC)	Onboard PTV entity that manages security alarms and emergency activities.	May be packaged with VLU or separately
PTV Manage Transit Signal Priority (PTVTSP)	Onboard PTV entity responsible for transit signal priority functions.	May be packaged with VLU or separately
PTV Manage VLU Data (PTVDAT)	This entity is the primary manager for onboard PTV data sharing. It provides a logging function, as well	

Table 4.2.3 PTV Logical & Physical Entities		
Name	**Typical Functions/Contents**	**Typical Variations**
	as a schedule, configuration, and software version management function.	
PTV Monitor/Report Adherence (PTVADH)	This entity performs the onboard PTV route and schedule adherence monitoring and reporting.	
PTV Monitor/Report Health (PTVHEL)	Onboard PTV entity responsible for monitoring & reporting PTV health and operational parameters.	
PTV Monitor/Report Location (PTVLOC)	Onboard PTV entity for monitoring & reporting PTV location.	
PTV Other Onboard Component (PTVOTH)	Onboard PTV component defined by an agency & authorized to participate in onboard dialog(s).	
PTV Passenger Information (PTVANN)	This entity provides automatic onboard sign updates, automated stop annunciation, and other passenger information functions.	May be packaged separately from the VLU
WLAN	▪ Provide access to fixed business systems when in range of a WLAN access point. ▪ Other entities	

4.2.4 Field Systems

Field Systems, similar to PTVs, exhibit different physical architectures from location to location with an agency. There are sometimes very substantial differences from one agency to another. The Model Architecture (shown in Figure 4.2.4) identifies four field-based logical entities:

- A Transit Employee Interface (TEI) used to communicate between transit employees and business systems. These devices may have many different physical attributes from office based PCs to hand-held wireless devices. TEIs are general purpose devices and may perform many non-TCIP functions (e.g. phone, email).
- A Roadside Fare Collection Entity (PTSFCFC) that manages field- (stoppoint/station) based fare equipment.
- A Roadside Security Entity (PTSFSEC) that manages field-based security devices (e.g. cameras, traveler alarms).
- A Roadside Traveler Information Entity (PTSFANN) that manages field- (stoppoint/station) based announcements and passenger information displays.

The Model Architecture makes no assertions about the packaging of these four entities. Thus these entities may be packaged together or separately, or incorporated into the devices they manage monitor or control. For discussion purposes the TEI is assumed to be packaged separately, and the other 3 entities are packaged in a Station/Stoppoint Logic Unit analogous to the onboard Vehicle Logic Unit.

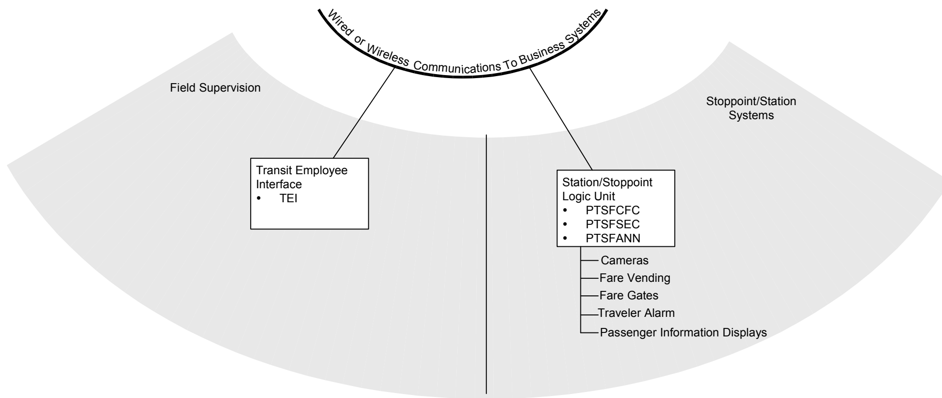


Figure 4.2.4
Transit Field Systems Model Architecture

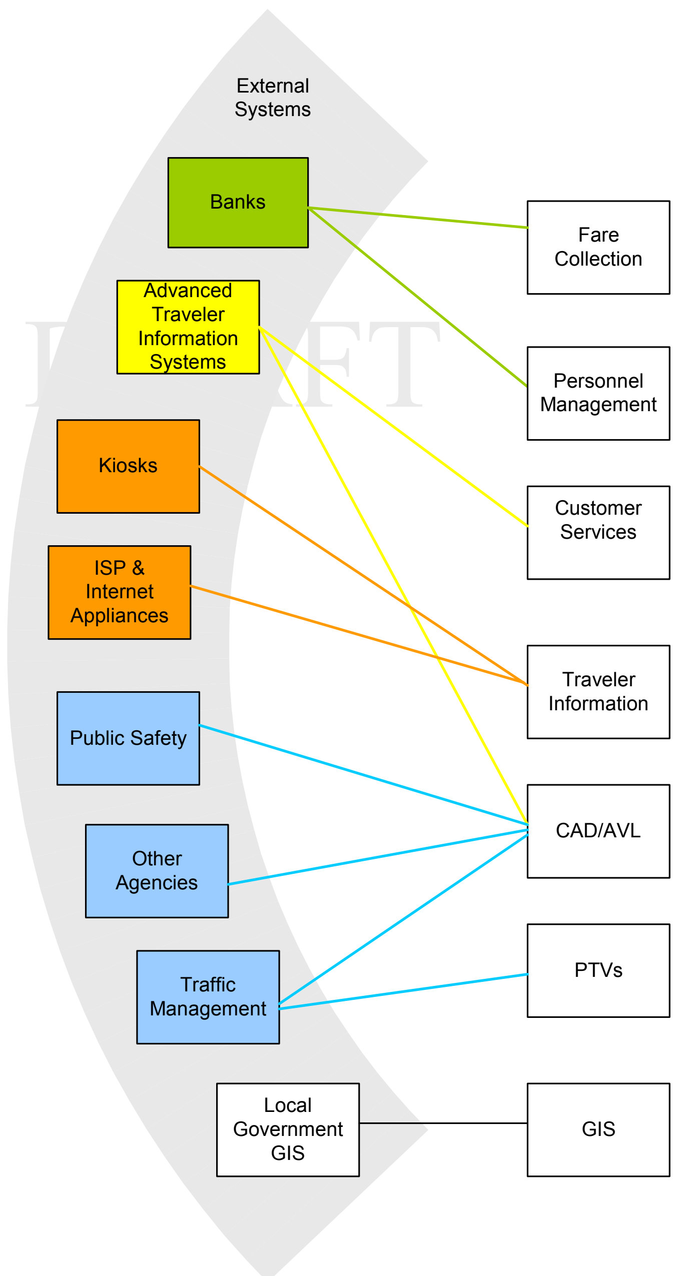


Figure 4.2.5
Transit Agency Interfaces to External Systems

4.2.6 Model Architecture Communications

The Model Architecture communications consists of 7 logical sub networks:

- Transit Data Network- used to communicate between agency business systems. Any agency network may be used, but for conceptual purposes, TCP/IP or UDP/IP communications over an agency-owned intranet is assumed. The TCIP communications over this network may be file transfers or dialogs using TCIP-XML messages.
- Business System to Field Systems – Consists of any combination of wide – or narrowband wired or wireless communications. May be used with TCIP-XML or narrowband-encoded messages. For conceptual purposes TCP/IP or UDP/IP communications are assumed.
- Business System to External Entities – May be any type of agency-to-external-entity network including leased lines frame relay or Internet connections. For conceptual purposes the Model Architecture assumes a TCP/IP or UDP/IP protocol with TCIP or non-TCIP XML messages, however in actual implementations any protocols and message types may be used (e.g. SNA, X.25).
- Business System to PTV Networks – this consist of two sub networks:
  - Wireless LAN – Provides wide bandwidth communications between the PTV and fixed business systems to allow loads/unloads of large data files. This sub network is assumed to be available only in limited geographical areas (e.g. vicinity of bus garage). For conceptual purposes TCP/IP or UDP/IP communications are assumed.
  - Wireless WAN – Provides lower bandwidth communications between the PTV and fixed business systems to allow ongoing communications while the PTV is away from the garage. This sub network may be an agency-owned private network using the TCIP-Polled protocol, or a Proprietary Communications Protocol, usually with narrowband-encoded TCIP messages. This sub network may also be a public wireless network used with TCIP XML or narrowband-encoded messages. This subs network may provide communications from the PTV to traffic management entities (roadside or office) to facilitate Transit Signal Priority.
- Onboard PTV Networks
  - Hub – This sub network connects onboard components with an Ethernet TCP/IP or UDP/IP connection. This allows onboard PTV components to exchange TCIP-XML messages
  - VAN – This sub network connects onboard components using SAE J1939 or SAE J1708 interfaces. TCIP messages are not exchanged on this sub network. Message definitions for this interface are in SAE J1587, J1708 and J 1939.

As with other parts of the Model Architecture, the Communications Network definitions above are non-normative. Transit agencies may adopt a wide variety of communications networks and architectures based on local needs and legacy systems. TCIP dialogs and file transfers will operate over many of these heterogeneous network environments.

4.2.7 Dialog Flow Conventions

In discussing the business processes within a transit agency, and the relationships between business processes and logical entities defined by the TCIP Model Architecture, it is frequently convenient to represent dialogs as flows between various entities. Dialog flows are similar to data flows in that they are intended to depict the movement of information between logical entities, however, data flows are normally defined as a unidirectional flow, whereas most dialogs involve bidirectional or multidirectional message transfers. To represent dialogs in a manner similar to data flows a convention for determining the primary direction of information flow within a dialog is required. Consequently table 4.2.7 defines the dialog flow directional conventions for TCIP.

The dialog flows shown in section 5 are representative of how the dialogs might be employed in an agency. The presence of a flow does not imply it is required and the absence of a flow does not imply it is precluded. In some cases there are alternative methods using the same flow and both may be illustrated on the same logical drawing, when in reality only one alternative would be implemented in an agency. In other cases alternatives are shown on separate logical drawings.

Table 4.2.7 Dialog Flow Direction Conventions		
Pattern	**Flow Source(s)**	**Flow Destination(s)**
Subscription	Publisher	Subscriber
Report	Reporter	Receiver
Command Response	Controller	Device
Silent Alarm	Operator	Dispatcher
Load	Fixed Component	Onboard or Field Component
Unload	Onboard or Field Component	Fixed Component
Voice Radio Call	Call Originator	Call Receiver
Signal Control & Prioritization	PRG	PRS
Blind Notification	Notifier	Receiver

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 National ITS 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 traveler information system, TCIP does not specify the screens, user interactions, etc. TCIP does provide the data structures and dialogs to facilitate the traveler information system obtaining schedule information from the scheduling system. TCIP also provides data structures and dialogs to allow one traveler information system to provide itinerary information to another (e.g. to another agency). TCIP uses extensible 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 traveler information system may be designated a subscriber, and a schedule repository a publisher 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 and planned business systems, legacy interfaces, as well as the TCIP interfaces. The agency architecture may specify a series of development 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 at the interface.

5.2 Security and Incident Management

The Security and Incident Management Business Process facilitates the provision of:

- Safe and secure environment for transit employees and customers
- A will thought out response to incidents affecting service and /or posing a potential hazard to transit employees or customers.

The Security and Incident Management Business Process Outputs are:

- Prevention of incidents to the extent feasible
- Prompt and effective detection of incidents that do occur
- Prompt and effective responses to incidents that are detected

The Security and Incident Management Business Process includes 5 stages:

- Planning
- Preparation
- Incident detection
- Incident response
- Incident follow-up

5.2.1 Security and Incident Management Planning Stage

The purpose of the planning stage is for the transit agency to determine the types of incidents, hazards, and threats that the agency faces, to assess the risks, and to develop and prioritize mitigation strategies. Agencies use a variety of manual, and automated tools and business systems to support the planning effort. Inputs to the process include historical incident information from the agency's operations, vulnerability assessments, agency infrastructure information, (including maps and drawings), and plans, procedures and policies of other local and regional public agencies, and plans for local or regional events (e.g. sporting, entertainment).

The planning process may be divided into a number of sub processes organized by the type of incident or for a single planned event. Planning for some incident types requires extensive coordination with external agencies and/or private companies. Generally this includes 4 types of planning sub processes:

- Planning for routine operationally disruptive incidents
- Planning for criminal incidents
- Planning for special events (e.g. sporting/entertainment/conventions)
- Planning for disasters and terrorist events

Planning for routine operationally disruptive incidents is generally based on past experience. These incidents include fires, police investigations, sick customers or employees, PTV breakdowns, traffic congestion, unplanned road closures etc. Joint planning with responders within or outside the agency may be required. Incident history information is useful in updating plans to compare existing plans and procedures with past incidents to verify that they cover the incidents occurring in the real-world.

Planning for criminal events is similar to routine operationally disruptive events in that coordination with transit and other police agencies is essential, and planning relies heavily on past events. The agency may also evaluate security features in this context and make plans to add security devices or to make infrastructure improvements. security devices include cameras, intrusion detection alarms access controls, etc. Infrastructure improvements may include closing areas to the public, increased lighting, etc.

Planning for Special Events is somewhat unique in that the specific event is known in advance. Transit schedules may be modified to run additional or special services, or patterns may be changed due to projected traffic patterns, street closures, etc. Alternatively, the schedule may not be modified, and preplanned detours may be implemented to accommodate anticipated conditions surrounding the event.

Disaster planning is at the opposite end of the spectrum from special events in that the timing and type of event are unknown in advance requiring planning for a broad spectrum of possible threats with random timing. Planning for these events tends to focus on unique transit requirements that may occur after such an event such as evacuation services, or movements of large groups of injured persons to hospitals or temporary care facilitates. Agencies may create and maintain special purpose schedules to implement in such events.

The primary requirement for interfacing to support incident planning is the ability to obtain transit information. TCIP supports incident planning information transfer needs with the dialogs as shown in Figure 5.2.1. Other planning documents, drawing, maps, agreements etc. are maintained in agency-specified formats and do not have TCIP dialogs.

- Infrastructure/Facility Information (A in Figure 5.2)
    - Subscribe Vehicle Inventory
    - Subscribe Stoppoint List
    - Subscribe Facilities
    - Subscribe Shelters (Stoppoint shelters not evacuation shelters)

- Incident History Information (B in Figure 5.2)
    - Subscribe Incident History

- Schedules (Normal of Incident Related) (C in Figure 5.2)
    - Subscribe Master Schedule Version
    - Subscribe Route Schedule
    - Subscribe Timepoints List
    - Subscribe Pattern List

- Geographical Information (D in Figure 5.2)
    - Subscribe Point Conversion
    - Subscribe Route GeoTrace

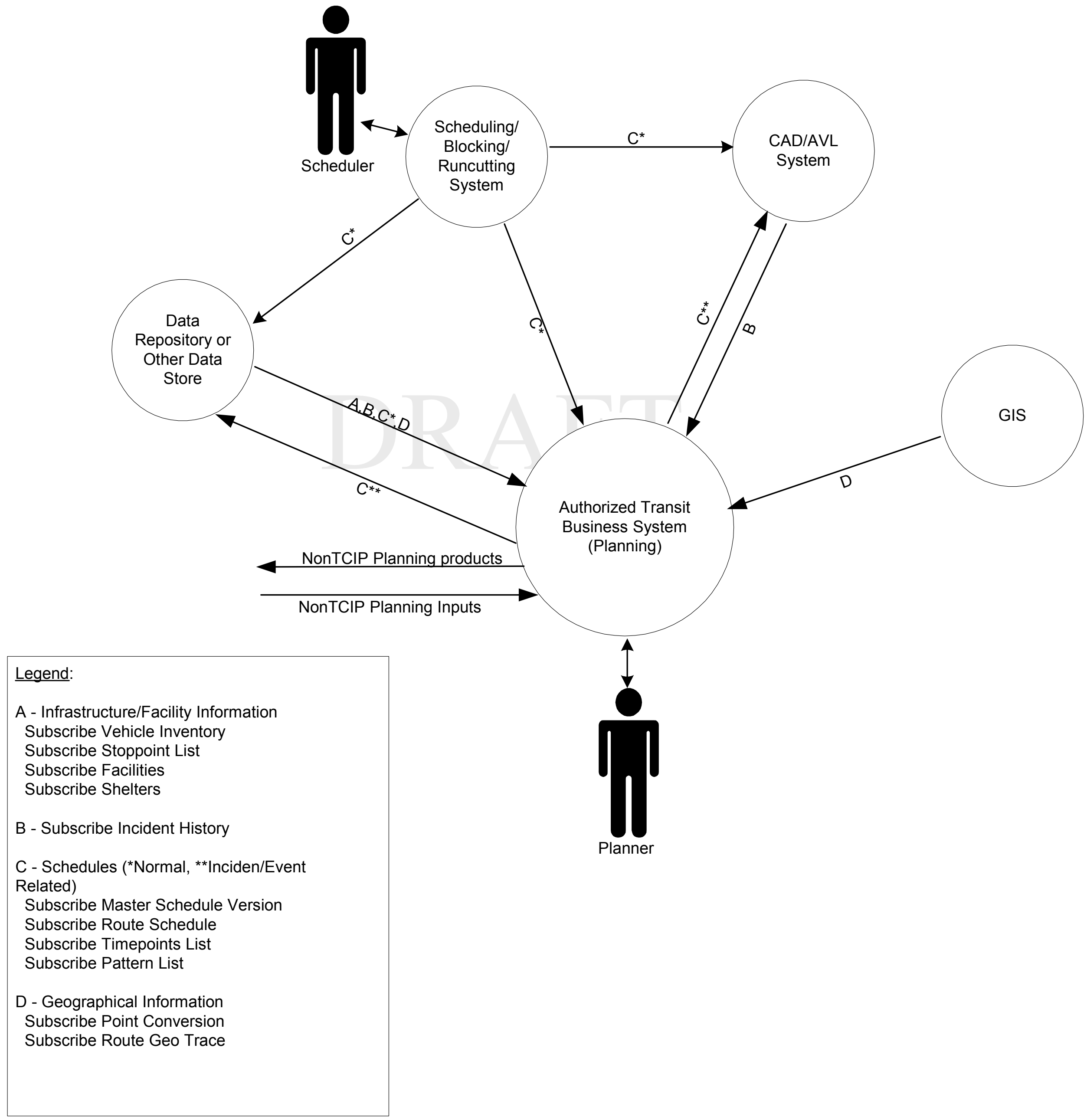


Figure 5.2.1
TCIP Dialog Support for Security and Incident Management Planning.

Figure 5.2.1 depicts the TCIP information transfers associated with incident planning. The Data Repository or other agency data store may provide infrastructure, past incident, schedule, and/or geographical information to an Authorized Business System used by planners. Alternatively, this information may be retrieved from other sources. The planners may modify schedules within the planning system(s), or alternative schedules may be developed using the Scheduling/Blocking/Runcutting System. Incident/Event Schedules may be maintained (stored) in the Scheduling/Blocking/Runcutting System, the Data Repository, or the CAD/AVL System.

5.2.2 Security and Incident Management Preparation Stage

The preparation stage is when the transit agency takes plans from the planning stage and implements actions to ensure the agency is ready to implement the plans. Activities in this stage include:

- Training - employees on plans and procedures which may include classroom training, drills, etc.
- Construction – including facility and infrastructure upgrades, hardening, installation of cameras, sensors and lighting, modification to PTVs etc.

No TCIP information transfer requirements have been identified for the preparation stage of the Security and Incident Management Business Process.

5.2.3 Incident Detection Stage

The prompt detection of incidents at the Transit Control Center is often one of the most critical factors in mounting an effective response. Detection occurs in a variety of ways including:

- Voice notifications by telephone or radio from an agency employee, customer or other agency.
- News feeds to the control center
- Data notifications from a variety of sources and methods
  - Incident Notifications from other agencies (transit, highway department, public safety, etc) governed by IEEE 1512
  - Traveler Alarms from stations/Stoppoints (TCIP Report Traveler Alarm dialog)
  - Passenger Alarms onboard PTVs (TCIP Report Passenger Alarm dialog)
  - Operator Alarms from onboard PTVs (TCIP Report Operator Alarm dialog)
  - Covert Alarms from onboard PTVs (TCIP Silent Alarm dialog)
  - Health Alarms from onboard PTVs (TCIP Subscribe PTV Health Alarms dialog)
  - Farebox Alarms from onboard PTVs (TCIP Subscribe Fare Collection Health dialog)
  - Farebox Collection Alarms from station/stoppoint based fare collection equipment.
  - Incident Notifications from agency employees (TCIP Report Incident dialog).
  - A text or canned message from a PTV operator.

Once the Control Center receives the notification, an incident report is created manually or automatically in the CAD/AVL System. Once an incident report is created, the incident moves to the response stage. Figure 5.2.3 depicts the information flows for incident detection.

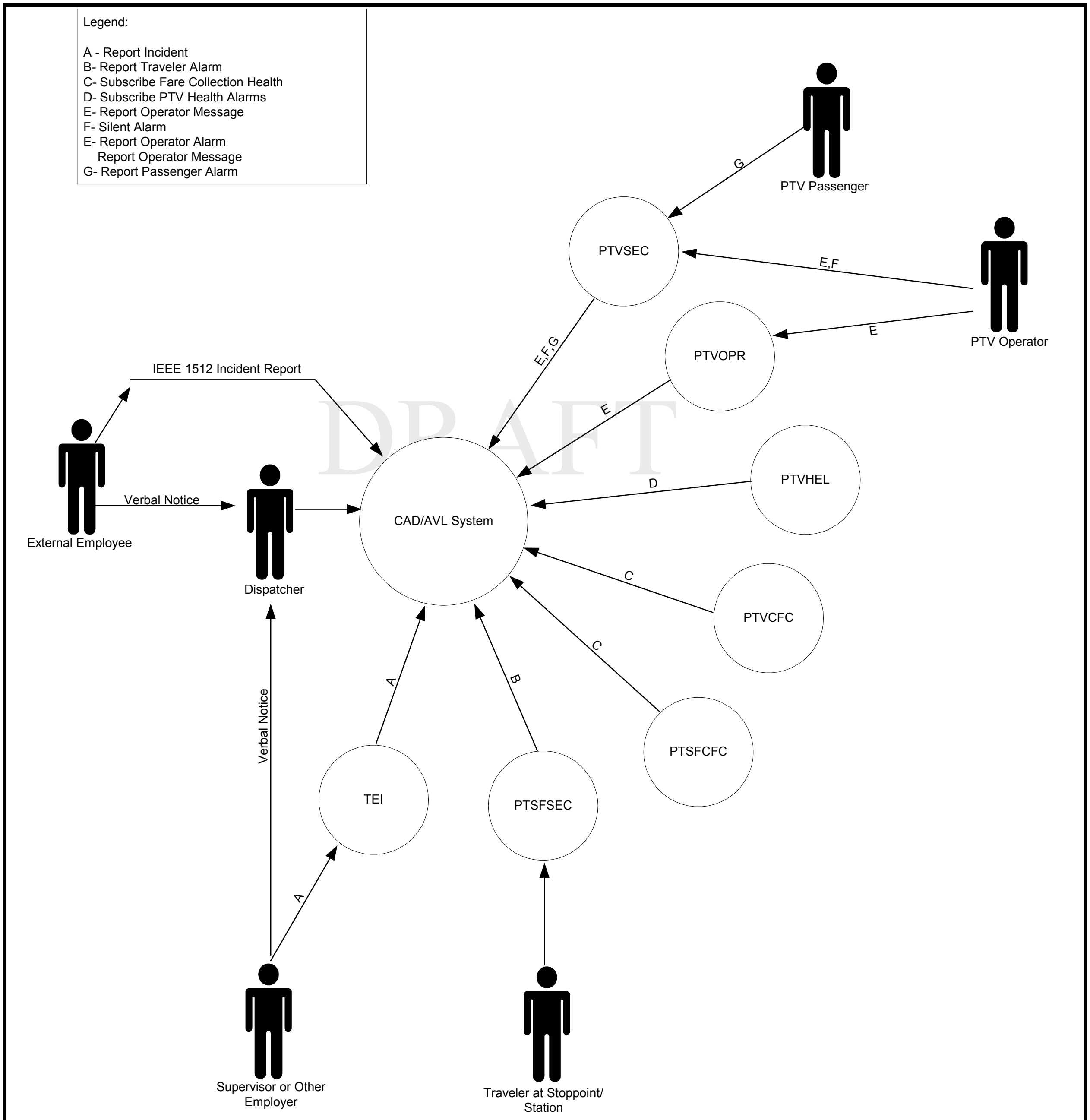


Figure 5.2.3
Incident Detection Dialog Flows

5.2.4 Security and Incident Response

Once the incident is reported to the Control Center, the response stage is executed. In this stage resources are mobilized to resolve, clear and close an incident. During this process, the Control Center may need to:

- Request/Provide assistance to/from external agencies
- Request/Provide status to/from external agencies
- View video images of events at the incident location
- Receive and distribute incident updates from/to transit employees
- Implement and cancel detours
- Exchange text messages with PTV Operator(s)

- Monitor covert audio from a stoppoint or PTV
- Clear/Cancel a silent alarm from a PTV
- Communicate by voice radio with PTV operators or other employees
- Change vehicle or operator assignments
- Dispatch agency responders to the incident (e.g. agency owned tow truck)
- Disable/Enable a PTV

Figure 5.2.4 illustrates the use of TCIP dialog in the Incident Response Stage, and subsequent subsections describe these processes.

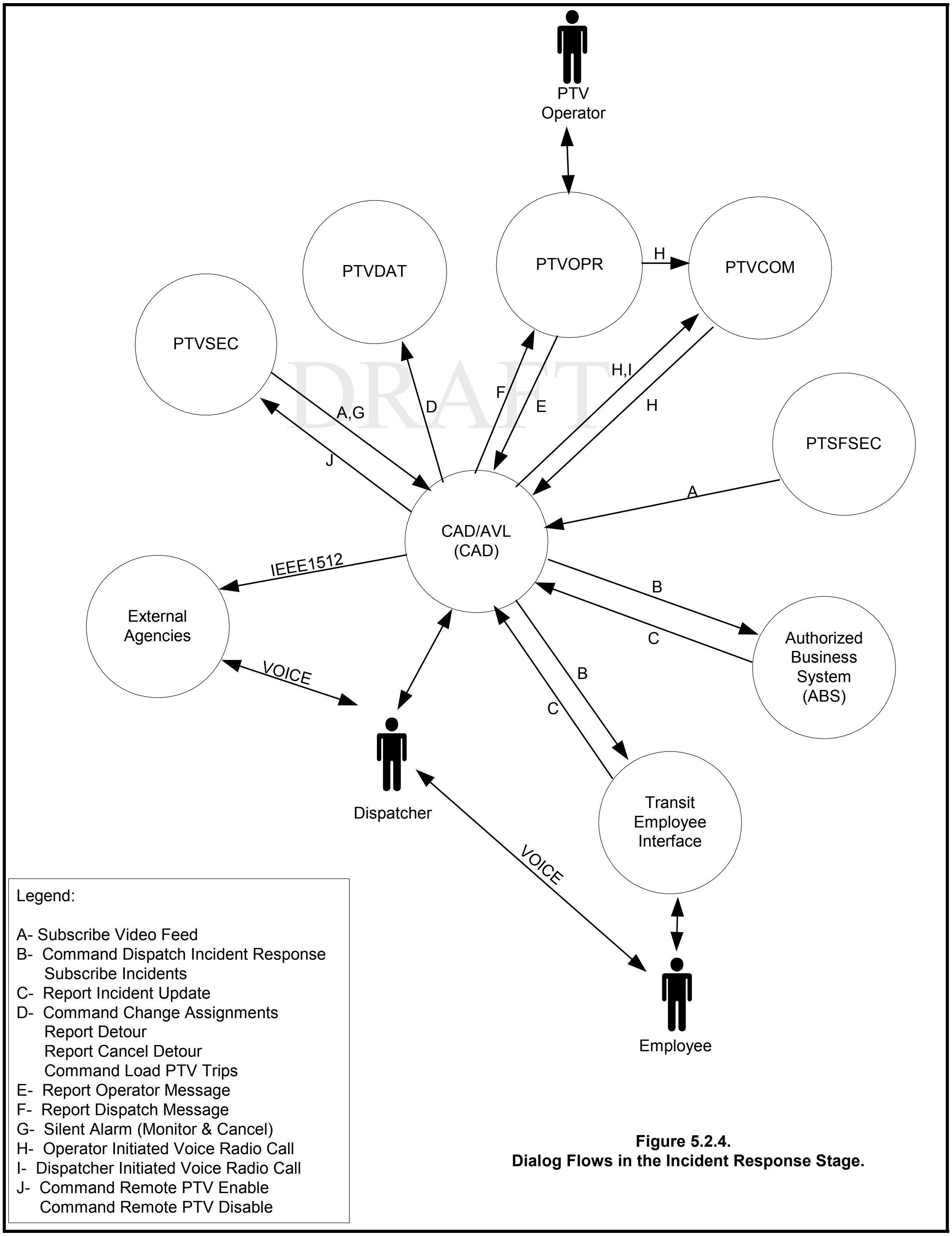


Figure 5.2.4.
Dialog Flows in the Incident Response Stage.

5.2.4.1 External Agencies

Interactions with external agencies may be by voice (radio or telephone) or by data. Voice communications are based on local or regionally developed plans and procedures. Standardized interagency data exchanges to manage incidents are defined by IEEE 1512.

5.2.4.2 View Video Images at the Incident Location

If the infrastructure (communications bandwidth and cameras) is in place, the dispatcher may be able to view near-real-time video imagery from the incident scene. It may be possible to present current images from a selected camera or past images (e.g. back up to the incidents occurrence) from a PTV or PTSF. The Subscribe Video Feed dialog supports this capability.

5.2.4.3 Receiving and Distributing Incident Updates

As the incident progresses, verbal and/or data updates to the incident arrive at the Control Center. The dispatcher enters the verbal update reports into the CAD/AVL System via the Report Incident Update dialog. The initial incident report and incident updates are distributed out to authorized subscribers using the "Subscribe Incidents" dialog.

5.2.4.4 Implement and Cancel Detours

Depending upon the type and nature of the incident, it may be necessary to detour a PTV (or all PTVs on a route) from their normal path. After the dispatcher enters the appropriate information into the CAD/AVL System, the Report Detour dialog transfers the detour to the PTV. The Report Detour dialog provides the following features:

- Ability to define way points along the detour
- Ability to add and delete timepoints and stoppoints to/from the trip
- Ability to change PTV's scheduled timepoint arrival times beyond the detour
- Ability to change the PTV's destination sign approaching and on the detour

When the detour is no longer required, the dispatcher enters the appropriate information into the CAD/AVL System and the Report Cancel Detour dialog conveys it to the impacted PTV(s).

5.2.4.5 Exchange Text Messages with PTV Operators

TCIP provides the capability for PTV Operators and Dispatchers to exchange canned and/or freeform text messages. This ability allows a dispatcher to provide instructions on how to respond to the incident to operators, and for operators to provide incident status or other information to the dispatcher without tying up the voice radio. These capabilities are provided by the Report Dispatcher Message and Report Operator Message dialogs.

5.2.4.6 Monitor and Cancel Silent Alarms

On appropriately equipped PTVs, the PTV operator can actuate a covert ("silent") alarm. This provides a notification to the dispatcher, and an acknowledgement back to the driver. In some agencies the acknowledgement is generated by the CAD/AVL System as soon as the alarm is received, on other agencies, the acknowledgement is withheld until the dispatcher indicates to the CAD/AVL System that he has seen the alarm. After the alarm is activated, the dispatcher maybe able to monitor events on the PTV using a covert microphone and an audio feed over the voice radio. At some point the operator indicates that the incident is over and cancels the alarm. The alarm remains active however, until the dispatcher confirms the incident closure.

5.2.4.7 Voice Communications Between Dispatchers and PTV Operators

In the course of an incident it is frequently necessary for the dispatcher and PTV operator to converse on the voice radio. On many agencies a shared voice/data radio is used. Voice Communications can be initiated by either the Dispatcher or the PTV operator, and the PTV Operator may indicate that the need to communicate is urgent. In any case the Dispatcher (through the CAD/AVL System) has the ultimate control over which PTV operators are allowed to enter into voice conversations.

These capabilities are provided by the "Operator-Initiated Voice Radio Call" dialog, "Dispatcher Initiated Voice Radio Call" dialog and the "TCIP Polled Protocol".

5.2.4.8 Change Vehicle or Operator Assignments

Incidents may force an agency to replan its operator and vehicle assignments or to change the trips assigned to vehicle/operator pair. The dispatcher enters these changes into the CAD/AVL System. The "Command Load PTV Trips" dialog and/or the "Command Change Assignments" dialog transfer the new instructions to the PTV.

5.2.4.9 Dispatch Agency Responders

The dispatcher upon determining that agency responders are required at an incident location requests or directs appropriate agency assets to the site. The "Command Dispatch Incident Response" dialog may be used to direct responders, or this may be done via voice communications, email, or using other agency-specified means.

5.2.4 10 Disable/Enable a PTV

In some severe circumstances (e.g. hijacking), a dispatcher may enter a command into the CAD/AVL System to disable an appropriately equipped PTV. The dialog "Command Remote PTV Disable" conveys the command to the PTV. When the dispatcher determines that the PTV should be enabled, and enters the appropriate information into the CAD/AVL System the"Command Remote PTV Enable" dialog conveys the enablement to the PTV. Enable/Disable commands require a key code to be accepted by the PTV.

5.2.5 Incident Follow Up

After an incident is cleared, additional follow-up efforts may be required. Depending upon the type of the incident it may be necessary to gather evidence for a criminal trial, and/or to recreate elements of the incident and the response to the incident as an input into future incident planning.

Video information is captured by cameras installed on PTVs and in PTSFs. This information is periodically unloaded to an archive or data store. During the follow up phase, these images may be retrieved for evaluation or use as evidence.

Business systems create log files of events as they transpire and these log files may provide useful information for recreating or evaluating incidents. Business system log files are not standardized by TCIP.

PTVs maintain logs of operating events while operating and this information is unloaded periodically to a data store or archive. During the follow up phase, these records may be retrieved for evaluation or used as evidence.

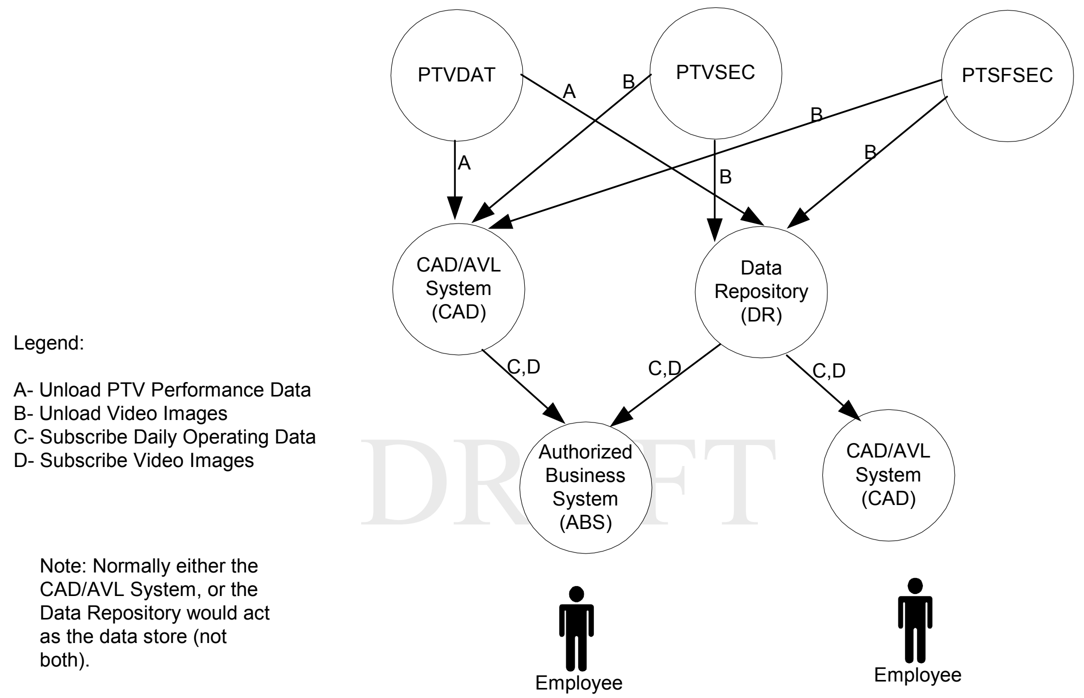


Figure 5.2.5
Incident Follow Up Dialog Flows

5.3 PTV Operations

The PTV Operations business process is the core of the transit business, and the most complex of the business process defined in TCIP. This process has four stages:

- Preparation for In-Service Operations
- Normal PTV operations
- Exceptions to Normal Operations
- Close-Out of PTV Operations

5.3.1 Preparation for In-Service Operations

Prior to entering service a variety of actions take place to ensure that a PTV is ready to enter service. Assume that the PTV has already been assigned a block of work for the day as described in section 5.7.1. The garage-based supervisor obtains a pull-out list from the Garage Operations System (GOS) or Transit Employee Interface (TEI). The pull-out list enumerates scheduled pull-outs and the operator, PTV, route and time for each.

The PTV operator is generally required to inspect the PTV prior to service. Some agencies also require a supervisor to inspect the PTV. If the PTV does not pass the pre-service inspection this must be reported to the appropriate business systems such as Maintenance Management (MM), the Garage Operations System (GOS), and the CAD/AVL System (CAD). Normally this will lead to a quick remedial action, or a change to the PTV block assignment.

Before, during, or after the inspection, the PTV operator starts the bus and the VLU and this is reported to the CAD/AVL System indicated to the dispatcher and logged. Since the VLU and the PTV engine may start at different times, multiple startup reports maybe generated. The PTV operator also signs on to the PTV, and similarly this is reported to the CAD/AVL System and logged. Agency specific policies determine the recovery mechanism if the wrong operator logs onto a PTV.

After the PTV and its computers are started, PTVCOM determines if the Wireless LAN is available, and if so notifies onboard entities so that data loading can begin. New data is only loaded if the current data is not already onboard. The following items are loaded as needed at this stage:

- Software (for any onboard components)
- Configuration Data (for any onboard components)
- Fare Data (to the onboard component that contains PTVCFC)
- Schedule Data (to PTVDAT)
- Stop and Destination Annunciation Data to support stop announcements and destination sign updates (to PTVANN)
- Canned Message Text to support efficient operator to dispatcher and dispatcher to operator messaging (to PTVOPR)
- Transit Signal Priority Business Rules to define interactions with equipped intersections. (Not all PTVs are TSP-capable)
- Alarm Limits and Other Parameters and Thresholds
- Background Map Images (Not all PTVs are map-display capable and agencies may limit the geographical scope of maps)

Any of these load processes can be aborted due to an invalid file, however the Schedule Data and Fare Data Loads have explicit dialogs for reporting validation errors on loaded files.

The garage-based supervisor has responsibility for ensuring that the PTV operator is fit for service (alert, sober, proper uniform, other agency specified criteria). If the PTV operator is not deemed fit for service, this will normally result in an operator assignment change. If the PTV and operator are deemed ready for service, the supervisor allows the PTV to leave the garage at the time designated on a pull out list, and reports the pullout using a TEI or via the Garage Operations System.

Figure 5.3.1 illustrates the data loading dialog flows during the preparation for in-service operations stage.

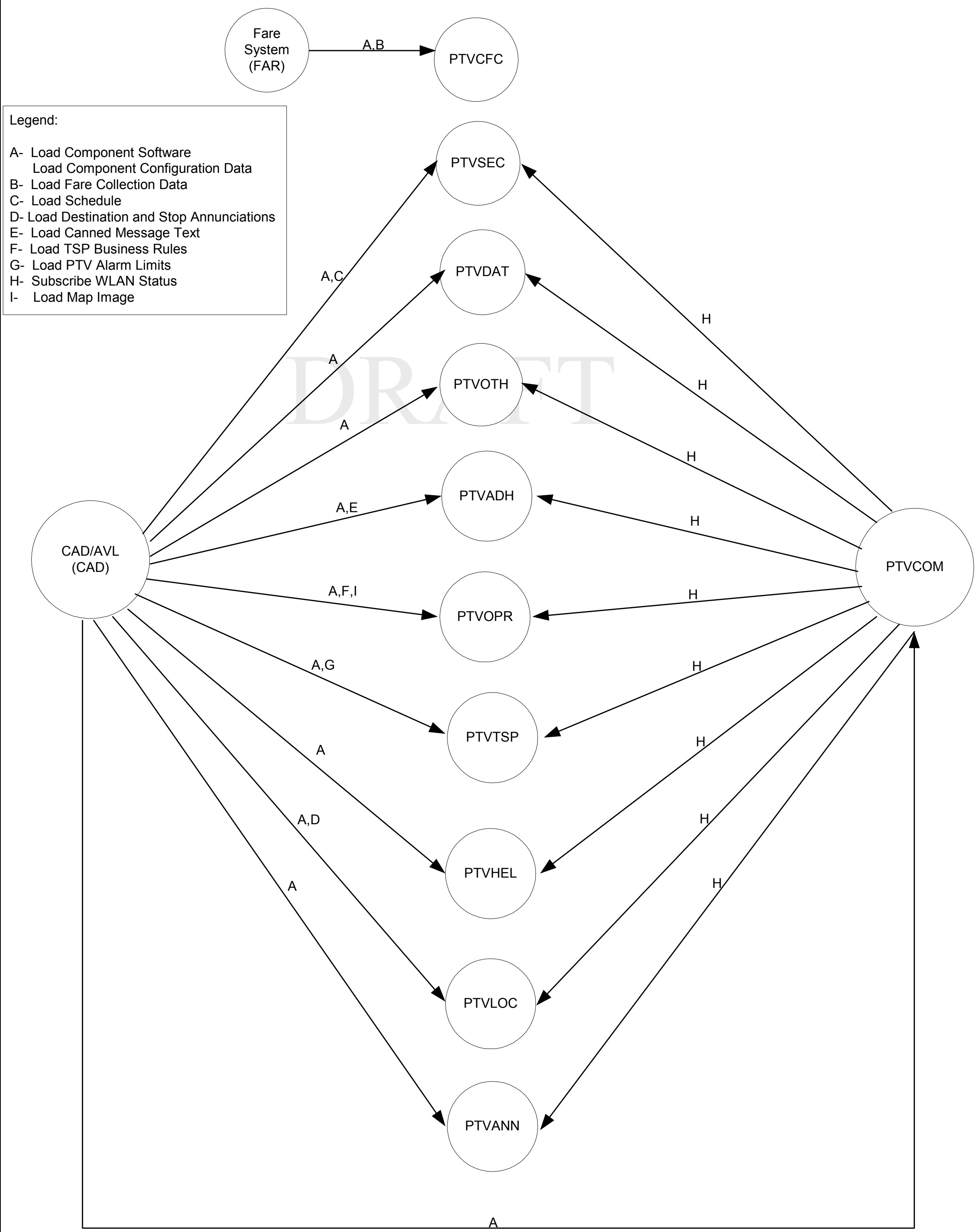


Figure 5.3.1.A
Data Loading During the Preparation for In-Service Operations Stage Without a Data Repository.

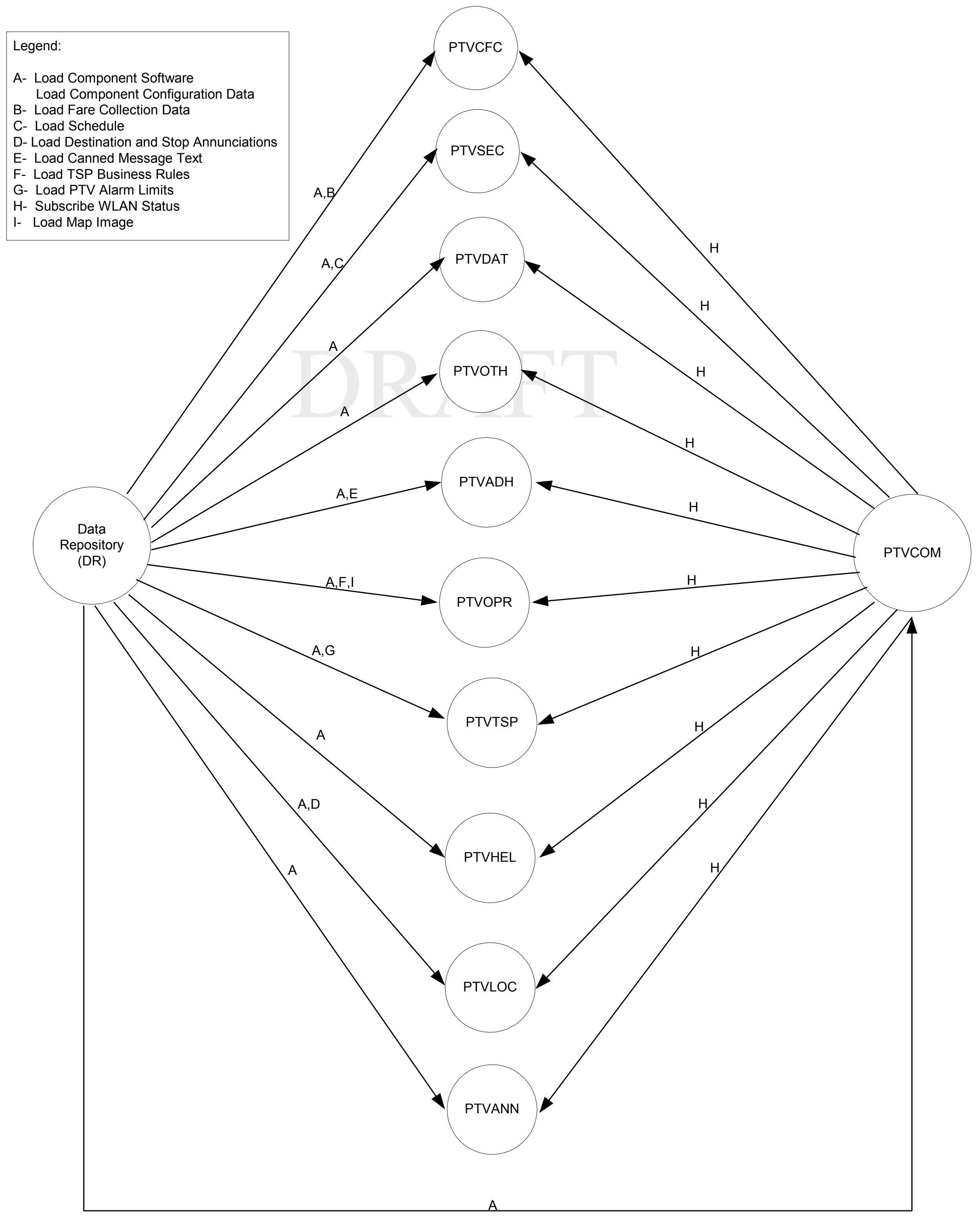


Figure 5.3.1.B
Data Loading During the Preparation for In-Service Operations Stage Without a Data Repository.

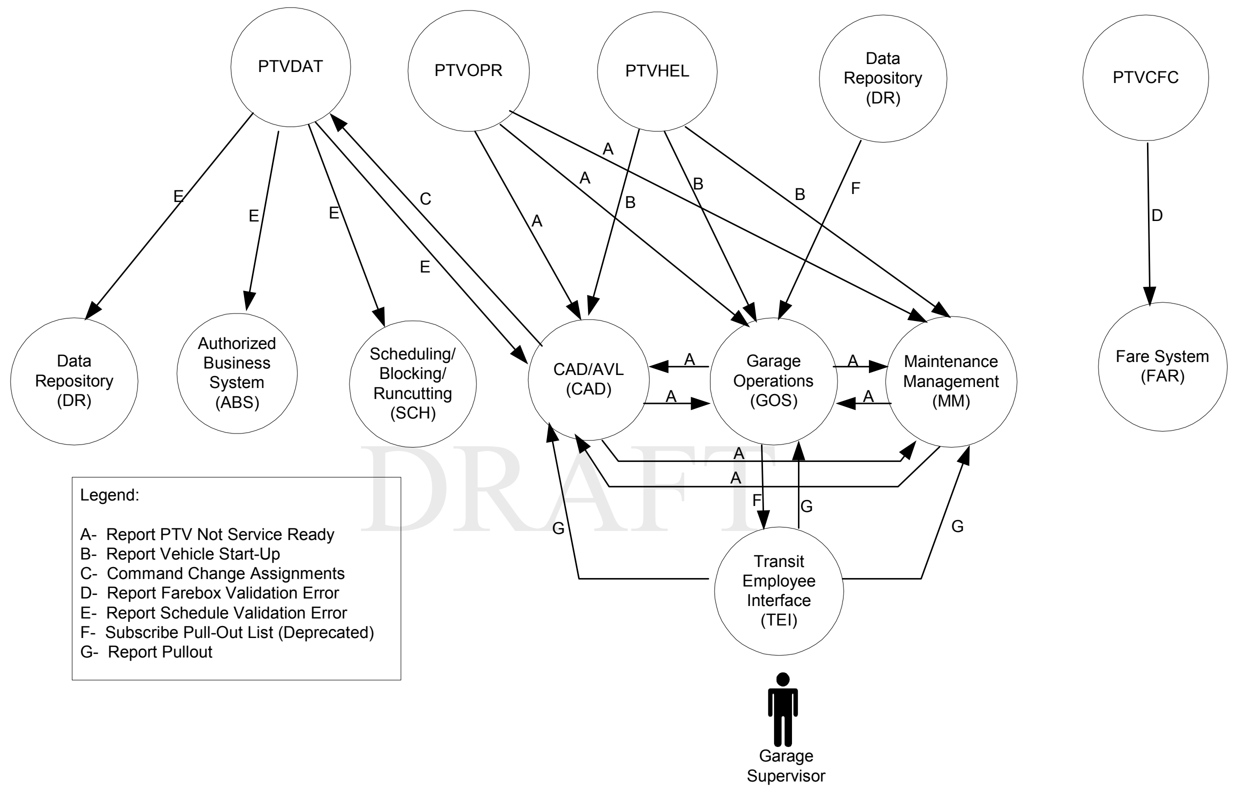


Figure 5.3.1.C
Other Dialog Flows During Preparation for In Service Operations Stage

5.3.2 Normal PTV Operations

PTV operations differ significantly between transit agencies . This section describes these operations for a generic agency an illustrated TCIP interactions associated with those operations

5.3.2.1 PTV Movements

After the PTV leaves the garage, its first trip is usually from the garage to an initial stoppoint. This is reflected by a scheduled out of service trip containing two timepoints at the garage and at the first stoppoint. This trip is followed by a second in-service trip that starts with the initial stoppoint followed by a timepoint coincident with the initial stoppoint. The timepoint time associated with this timepoint reflects the scheduled departure time from the initial stoppoint. The PTV waits (usually with doors open) at the initial stoppoint until the scheduled departure time specified for the first timepoint.

Agencies may elect to include or omit stoppoint from the pattern segments that constitute the remainder of the revenue trip, however enroute timepoints are needed to establish schedule and route adherence.

If the trip contains a layover (PTV is scheduled to stop and wait for an interval at a stoppoint enroute), this is reflected by a coincident timepoint-stoppoint-timepoint sequence in the pattern for the trip. The timepoint time associated with the first timepoint reflects the scheduled arrival at the stoppoint, the timepoint time associated with the second timepoint indicates the scheduled departure from the layover stoppoint. This construct may be used to ensure PTVs remain at a stoppoint for a sufficient duration to allow transfers.

At the end of an in-service trip, the PTV may be scheduled for an out of service trip back to the garage, and out of service trip to a designated parking area (e.g. for an operator break), an out of service trip to another agency designated location, an out of service trip to the initial stoppoint for the next in service trip, or may immediately be followed by another in-service trip.

Scheduled out of service trips following an in-service trip are similar to the out of service trip to the initial stoppoint. The trip contains a timepoint indicating the start location and time for the out of service trip. If the bus is to go to a location and park (e.g. for a break), the parking location is reflected by two sequential timepoints at the parking location. The timepoint times associated with the two timepoints at the parking location indicate the scheduled arrival and departure times at that location. A final timepoint in the out of service trip's pattern indicates the endpoint for the out of service trip. If the out of service trip is back to the garage at the end of the PTV's work block, it contains a timepoint for where the vehicle left service and a timepoint for the pull-in location. The timepoint times for these timepoints indicate the scheduled out of service and pull in times.

5.3.2.2 PTV Location Tracking

The current location of PTV's is of interest to a variety of transit entities including onboard entities, the CAD/AVL System, and other agency designated Authorized Business Systems (ABS).

The PTV's raw location heading and speed is determined by the PTVLDS entity (usually a GPS receiver). This entity reports raw location either directly to the Vehicle Logic Unit PTVLOC entity, or over the Vehicle Area Network (VAN). If the LDS reports directly to the VLU, then PTVLOC relays the information over the VAN to non-TCIP components. PTVLOC also obtains adherence information from PTVADH, and schedule information from PTVDAT and provides location data to any other onboard subscriber entities via the"Subscribe Onboard Location" dialog.

PTV Tracking by business systems can be implemented in several ways depending on the agency architecture. For agencies that use the TCIP Polling Protocol, PTVCOM obtains location, adherence and other day and provides it to the TCIP Polling Controller in response to each poll. The TCIP Polling Controller provides this information to the CAD/AVL System using the "Notify PTV Polling Result" dialog. Other business systems, then obtain PTV location from the CAD/AVL System using the "Subscribe Fleet Location" dialog.

For agencies that use public wireless networks, PTV location is provided to the CAD/AVL System using the "Subscribe PTV AVL" dialog. This dialog may also be used by other agency Authorized Business Systems to directly subscribe to PTV location, however, a more efficient use of wireless network capacity is for such systems to obtain PTV location from the CAD/AVL System using the "Subscribe Fleet Location" dialog.

Figure 5.3.2.2 illustrates the distribution of PTV location.

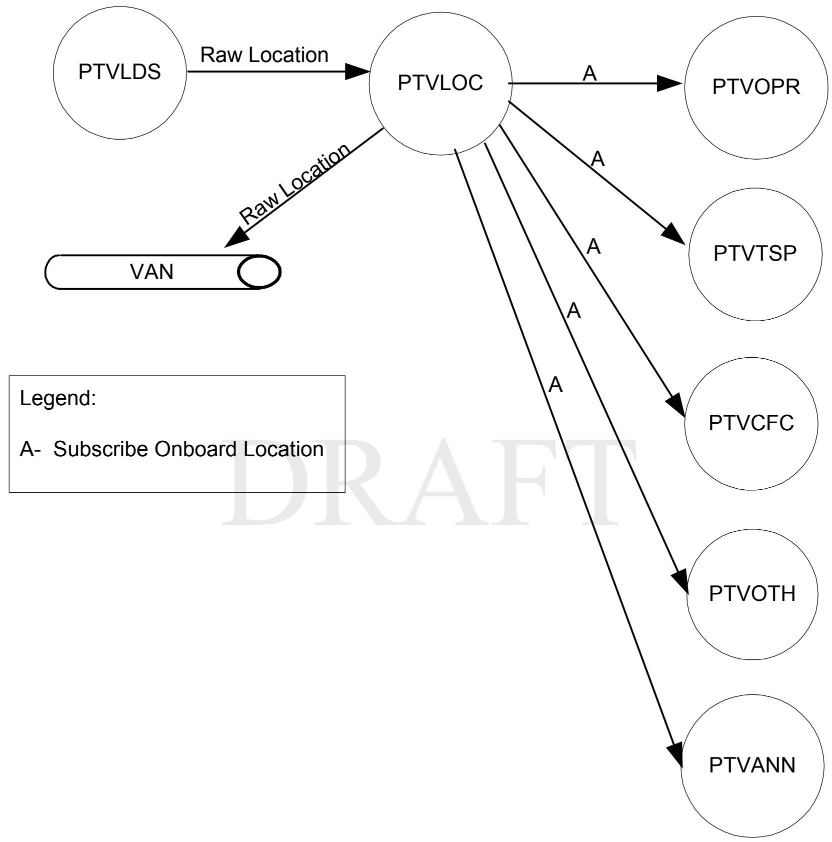


Figure 5.3.2.2.A
Onboard Distribution of PTV location with PTVLDS Directly Connected to VLU (PTVLOC)

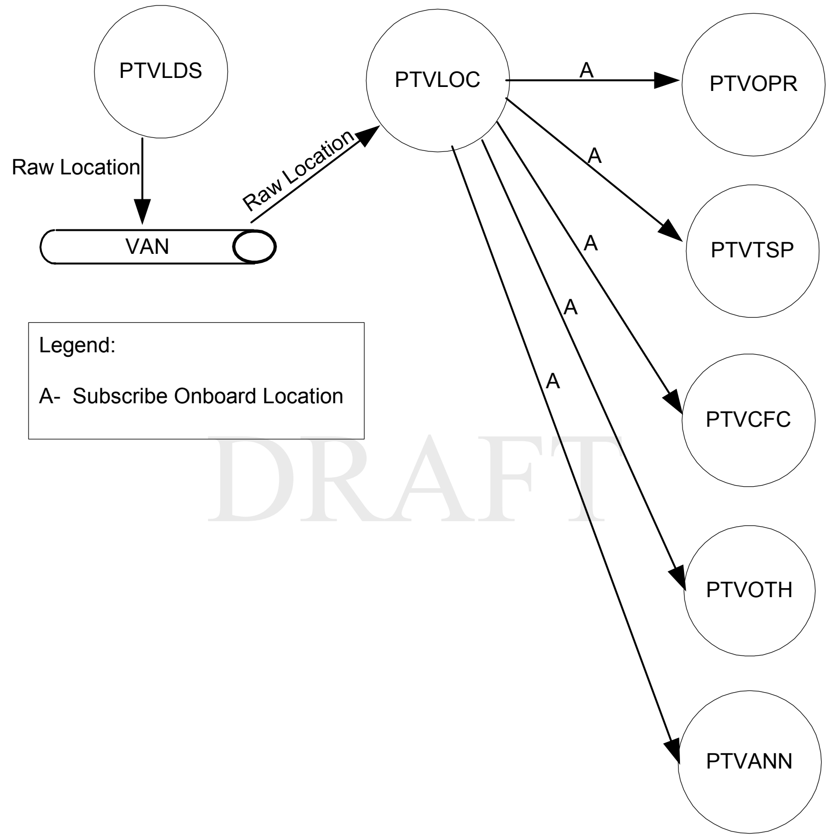


Figure 5.3.2.2.B
Onboard Distribution of PTV location with PTVLDS Directly Connected to VAN

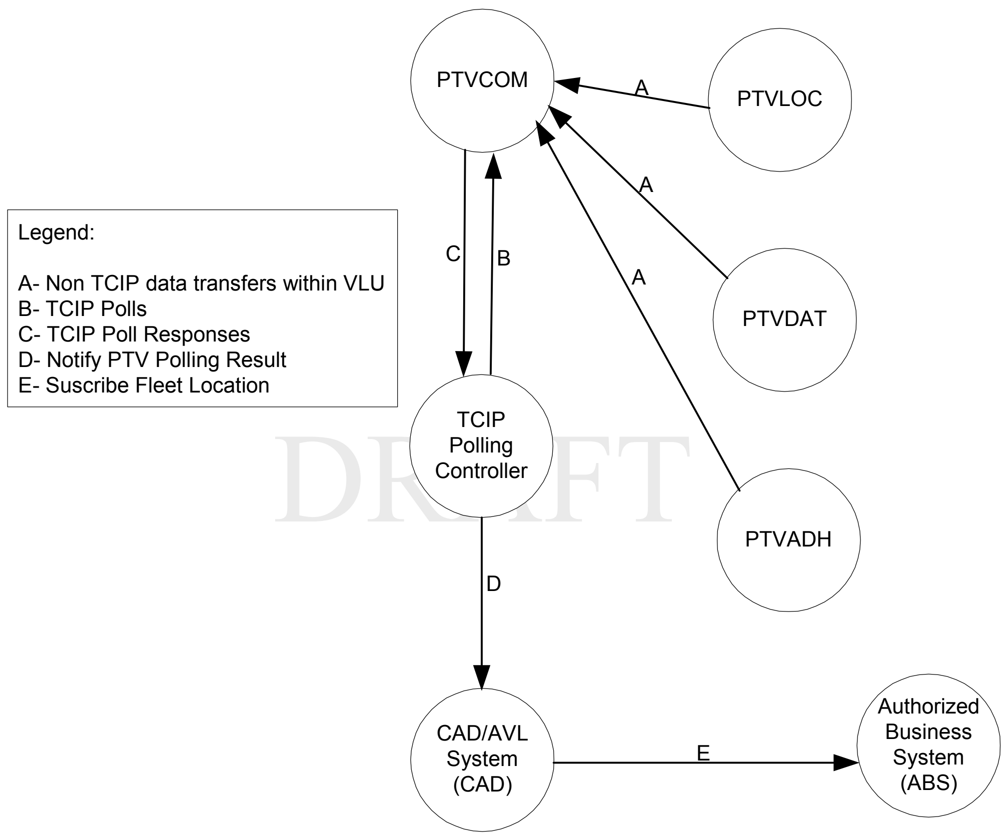


Figure 5.3.2.2.C
Distribution of PTV Location to Business Systems Using TCIP Polled Protocol.

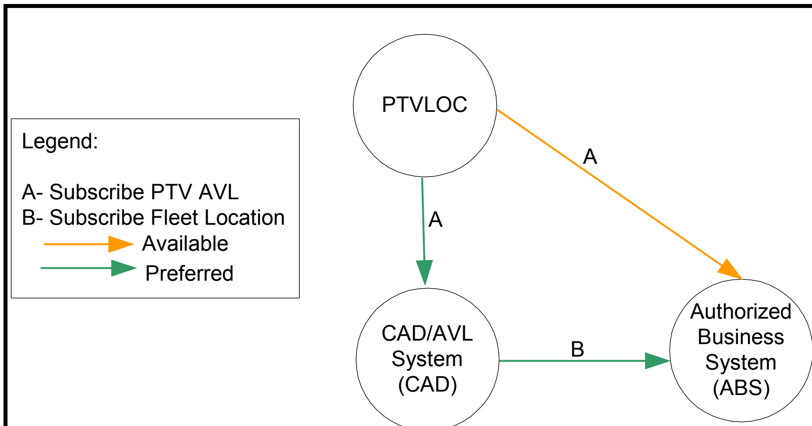


Figure 5.3.2.2.D
Distribution of PTV Location to Business Systems Using Public Wireless Networks.

5.3.2.3 PTV Operator Changes

From time to time agencies find it necessary to change PTV operators while the bus is enroute. These changes may be planned or unplanned. Planned (scheduled) changes may occur between trips or at a timepoint within a scheduled trip.

In the case of a scheduled operator change that occurs at the end of a trip, the vehicle assignment will contain a subsequent trip assigned to the PTV, but the operator assignment for the subsequent trip will have a different operator identifier.

If the operator change occurs mid-trip the pattern for the trip will include a layover (successive coincident timepoints for arrival and departure from a location). The old operator assignment will indicate that it ends at that timepoint identifier and the "new" operator assignment will indicate that it begins at that timepoint identifier.

After arriving at the location where the operator change is scheduled, the "old" operator signs off and this is reported to the CAD/AVL System (and optionally to the dispatcher). Similarly the "new" operator signs on and this event is also reported to the CAD/AVL System, and optionally the dispatcher.

If the operator change is unscheduled, the process of signing off and on is identical, however the notification to the PTV is different. In this case, PTVDAT already has an operator assignment based on the earlier data load. This assignment is changed by the dispatcher in the CAD/AVL System and communicated to PTVDAT using the Command Change Assignments dialog.

The dialog flows for PTV operator changes are illustrated in figure 5.3.2.3.A.

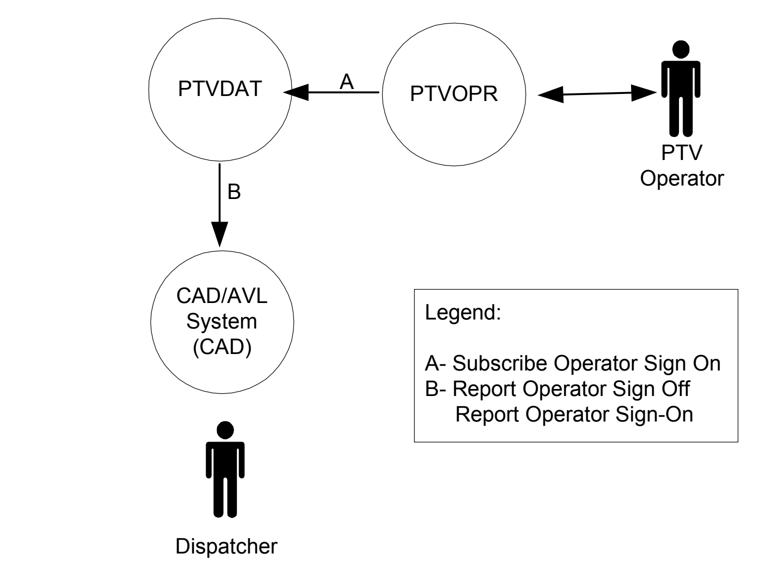


Figure 5.3.2.3.A
Dialog Flows for Scheduled PTV Operator Change

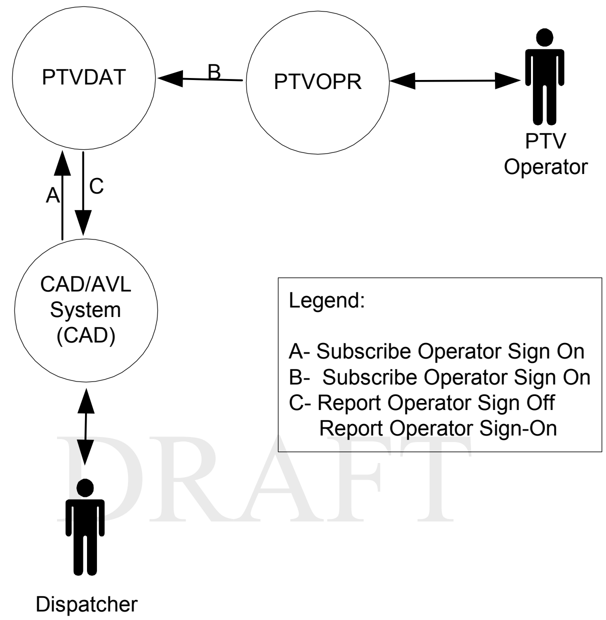


Figure 5.3.2.3.B
Dialog Flows for Unscheduled PTV Operator Change

5.3.2.4 Enroute Communications wit Dispatcher

PTVs often operate for extended intervals without requiring any interaction with dispatchers beyond Automatic Vehicle Location monitoring as described in section 5.3.2.1. There are however occasions where communications by voice or data are required.

The two major methods for dispatcher-operator voice communications are via radio or cell phone. PTV Operator Cell Phones are outside the scope of TCIP. Radio conversations may be initiated by the PTV operator or by the Dispatcher. The dispatcher-initiated call can be to the operator or to the annunciator (PTVANN) to make a remote announcement. Operator calls must be accepted by the dispatcher prior to being set up by the radio.

Enroute data communications between the operator and dispatcher are possible in either direction. These communications can use prepackaged (canned) messages, which transmit message numbers in a fill in the blank format with blanks filled by entries from numbered lists, this mechanism provides a wireless network-efficient means of text exchange for conversations that are foreseeable. An example use of this capability would be to tell a driver to stop picking up passengers until a specified stoppoint is reached. This type of action might be taken in response to a bus bunching.

Freeform text communications between the driver and dispatcher are also feasible. These communications require the extra effort to type the message and are not as network-efficient as canned messages, but have the advantage of allowing completely ad-hoc messages to be exchanged.

Figure 5.3.2.4 illustrates the TCIP dialog flows for enroute dispatcher-PTV communications.

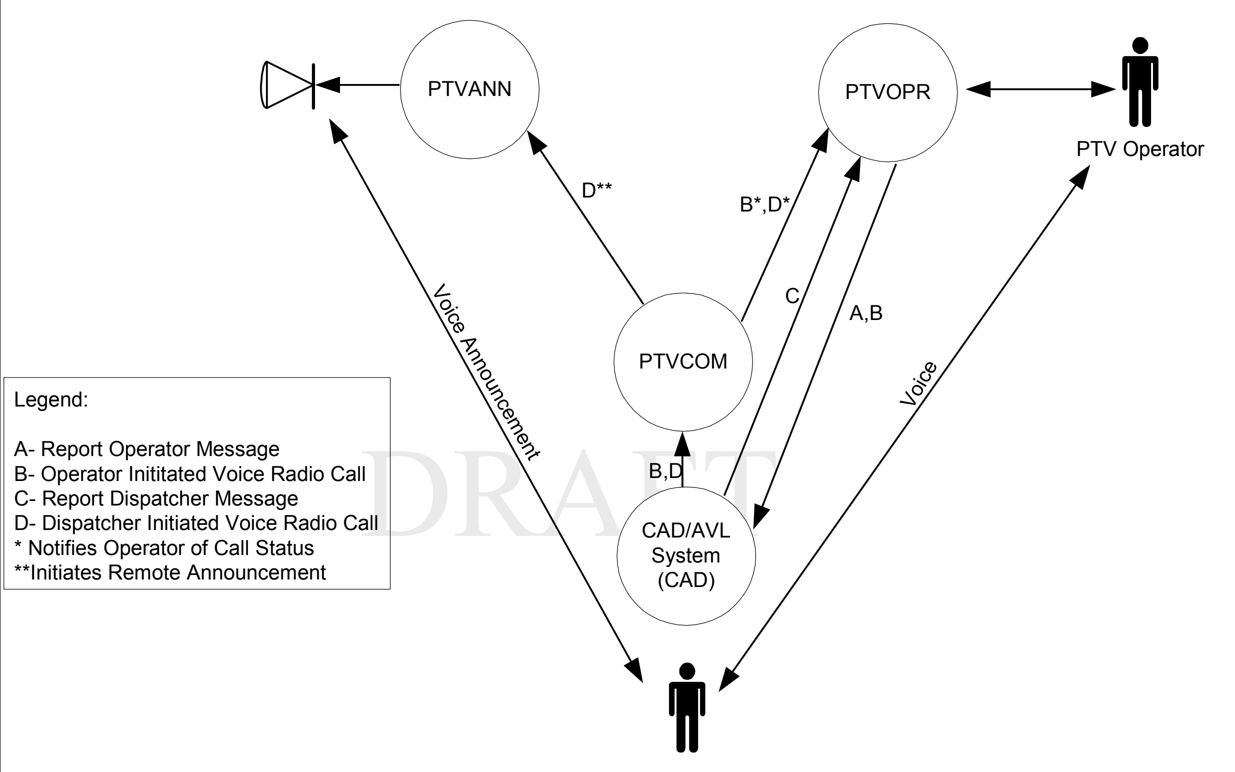


Figure 5.3.2.4
Dialog Flows for Enroute PTV-Dispatcher Communications

5.3.2.4 PTV Health and Parameter Monitoring

As the PTV progresses enroute, it is necessary to verify that the PTV and its onboard systems are operating normally. TCIP provides health alarm reporting dialogs for the PTV (as a whole) and for the farebox. The TCIP Polling Protocol also provides a mechanism to allow active alarms to be reported to the TCIP Polling Controller in response to a poll.

The default criteria for alarm generation (parameter ranges) are provided by the load PTV alarm limits dialog executed in the preparation for in-service operations stage. These limits can be overridden by individual subscription requests for health alarms.

TCIP also provides a mechanism to subscribe to onboard parameters (e.g. water temperature, oil pressure) that are broadcast on the VAN. Normally there will be no active subscriptions to these parameters, however, a subscription can be initiated by a business system in response to an alarm.

Figure 5.3.2.5 illustrates Enroute PTV Health and Parameter Monitoring.

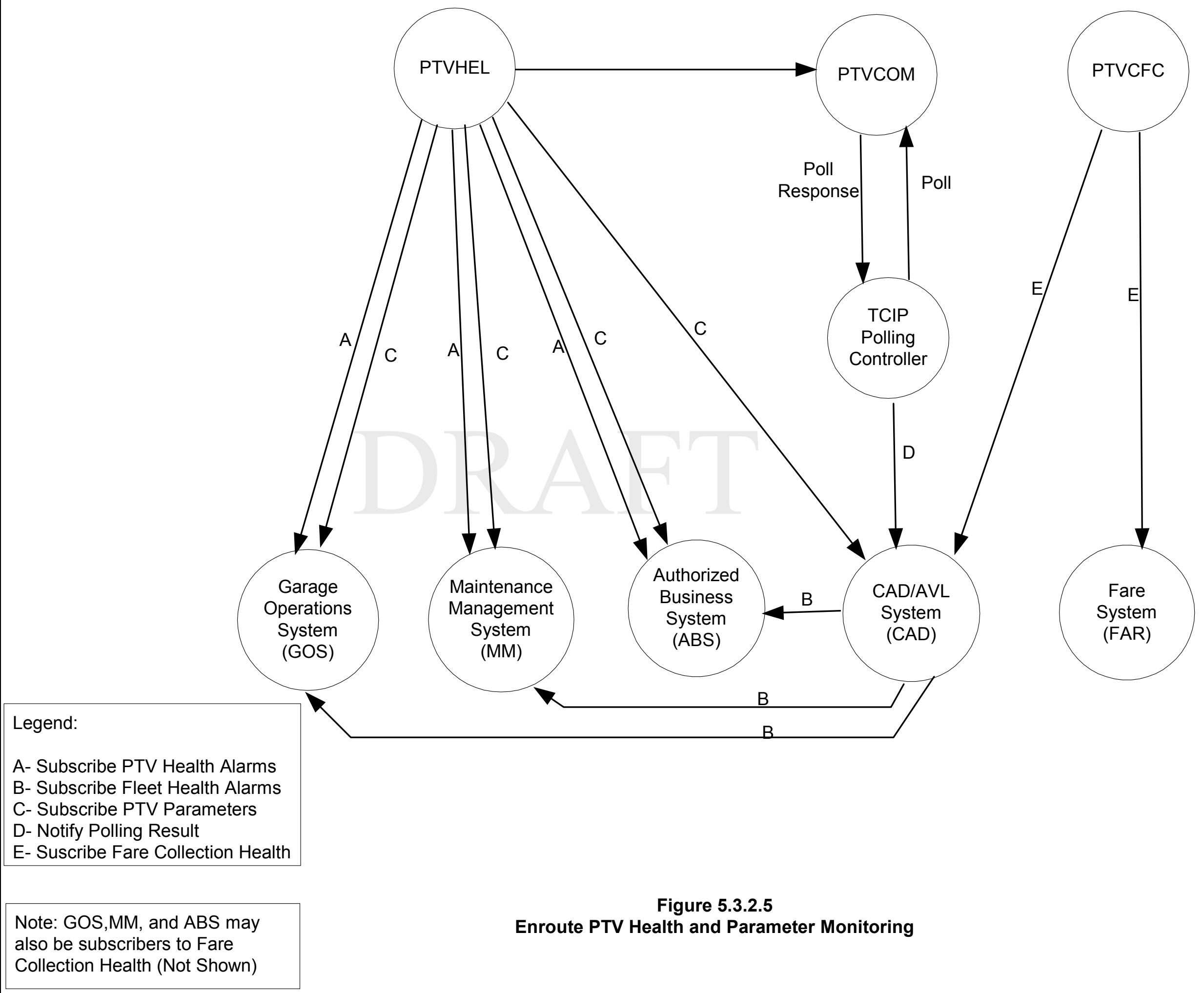


Figure 5.3.2.5
Enroute PTV Health and Parameter Monitoring

5.3.2.6 Transit Signal Priority (TSP)

As the PTV operates on its trips it may encounter intersections that are equipped to provide priority treatment to PTVs (e.g. early green, extended green, phase rotation) to allow the PTV to operate more efficiently and/or maintain schedule. Equipped intersections, and agreeing on acceptable strategies for TSP requires extensive coordination between transit agencies, traffic management, and traffic engineering. Although a PRG may request priority treatment, the traffic management system is not obliged to , and may not grant it.

All intersections are not equipped identically. NTCIP 1211 defines four scenarios for TSP operation. In addition TCIP recognizes an optical variation on Scenario #4, and an additional Scenario #5. These scenarios vary according to the physical location of the Priority Request Generator, and the path by which the priority request and its precursor information flows toward the Priority Request Server.

Figure 5.3.2.6 provides a conceptual overview of the entire TCIP process, independent of scenario numbers. Table 5.3.2.6 Summarized the various scenarios. Note that this figure includes the loading/unloading of information whereas the dialog flows only over enroute operations. Loading dialog flows are included in section 5.3.1 and unloading dialog flows are included in section 5.3.4. Accordingly, the remainder of this section assumes that any required loading of information to the PTV was conducted during the preparation (pre-departure) stage.

Note that network and server may cause priority requests to become “stale” before they can be processed. Implementing agencies should closely consider latency issues when implementing TSP.

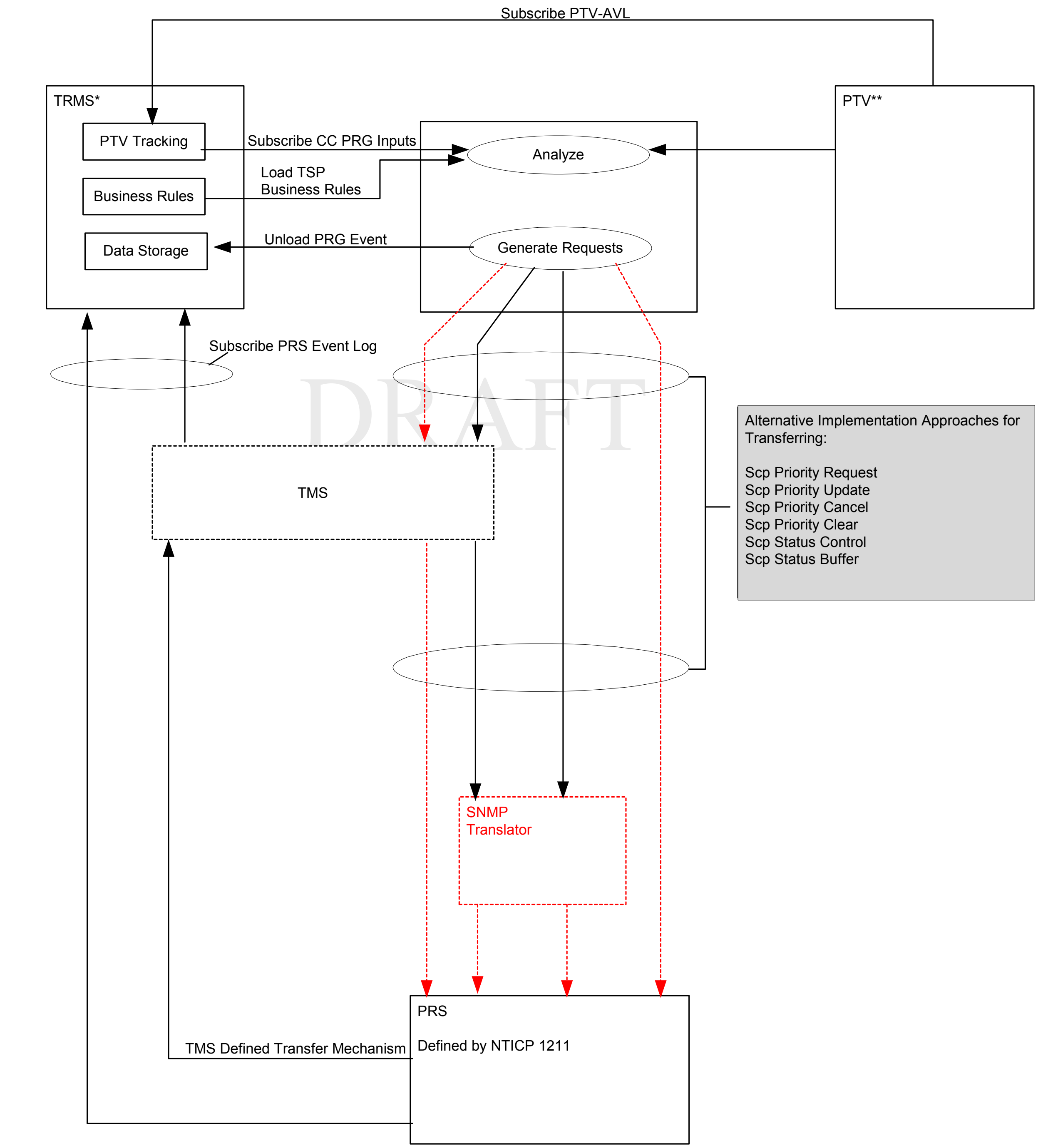


Figure 5.3.2.6
Scenario Independent Overview of Transit Signal Priority

Legend:
- → TCIP XML or Narrowband Message
- → TCIP/NTCIP 1211 SNMP Message
- * May be implemented by Data Repository and/or CAD/AVL.
- ** Contains multiple internal logical entities

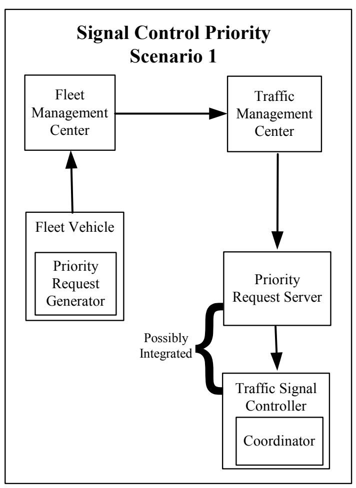
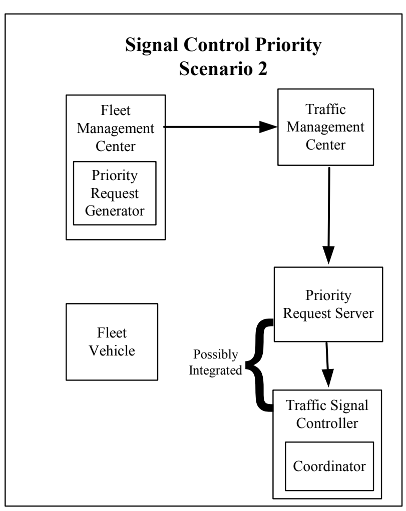
5.3.2.6.1 Priority Request Scenario 1

The first priority request scenario is specified by NTCIP 1211 to include an onboard PTV Priority Request Generator (PRG). The request is generated onboard and transferred from the PTV to the CAD/AVL System, onward to the Traffic Management Center and then to the Priority Request Server. This is depicted by figure 5.3.2.6.1.

In this scenario, PTVTSP must have been loaded with the”Load TSP Business Rules” dialog. PTVTSP receives real-time input from PTVDAT via the Subscribe Onboard PRG Inputs dialog which enables it to detect the PTV’s approach to an intersection, select a request strategy based on the business rules, and to send a signal priority request to the CAD/AVL System to be forwarded via the TMS to the PRS.

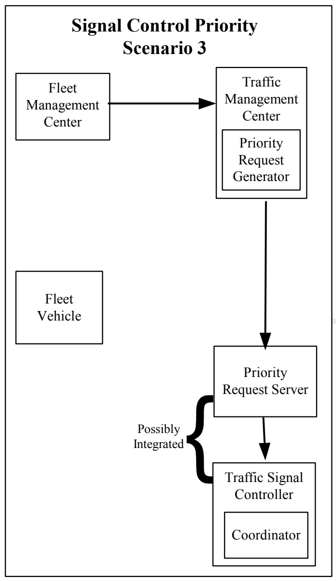
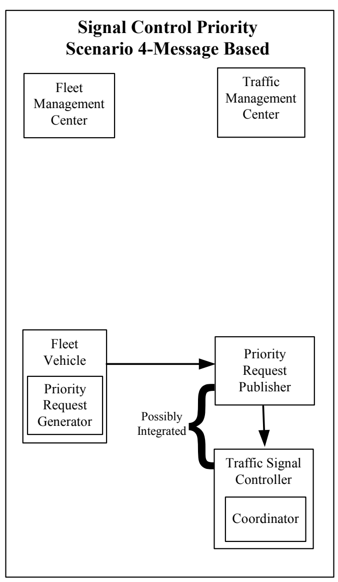
Implementation of this scenario requires close attention, by the transit agency as well as the traffic management agency, to ensure that cumulative network and server latencies do not delay the priority request excessively.

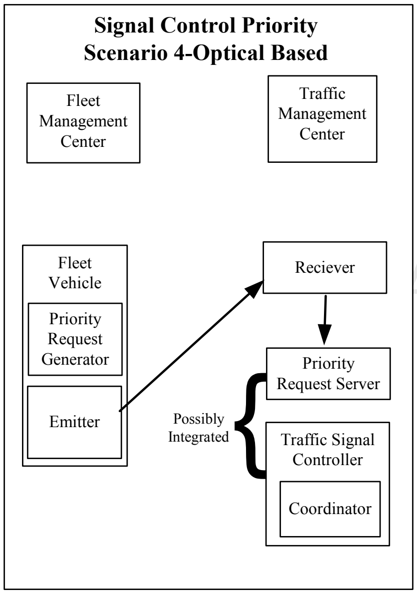
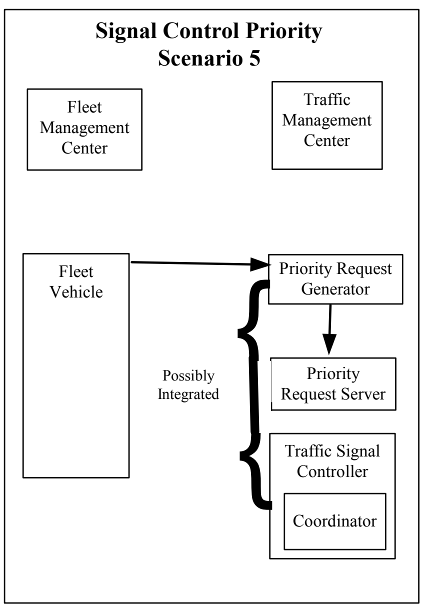
Note that the Scp Priority Request Scenario 1 dialog uses SNMP-encoded messages.

<table>
<tr><th colspan="3">Signal Priority Scenarios</th></tr>
<tr><th colspan="3">Table 5.3.2.6</th></tr>
<tr><th>Scenario</th><th>Figure</th><th>Description</th></tr>
<tr><td>1</td><td>
</td><td>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.</td></tr>
<tr><td>2</td><td>
</td><td>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.</td></tr>
</table>

Signal Priority Scenarios

Table 5.3.2.6

Scenario	Figure	Description
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.
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. This equates to NTCIP Scenario #4.

<table>
<tr><th colspan="3">Signal Priority Scenarios</th></tr>
<tr><th colspan="3">Table 5.3.2.6</th></tr>
<tr><th>Scenario</th><th>Figure</th><th>Description</th></tr>
<tr><td>4-Optical</td><td>
</td><td>The transit fleet vehicle carries a Priority Request Generator (PRG). Priority Requests are initiated via an optical link from the vehicle to the intersection. Neither the Transit Control Center, nor the Traffic Management Center are directly involved in the processing of priority requests or responses.</td></tr>
<tr><td>5</td><td>
</td><td>The transit fleet vehicle conveys inputs to the request generation process to a roadside-based PRG. Neither the Transit Control Center not the Traffic Management Center are directly involved in the processing of requests or responses.</td></tr>
</table>

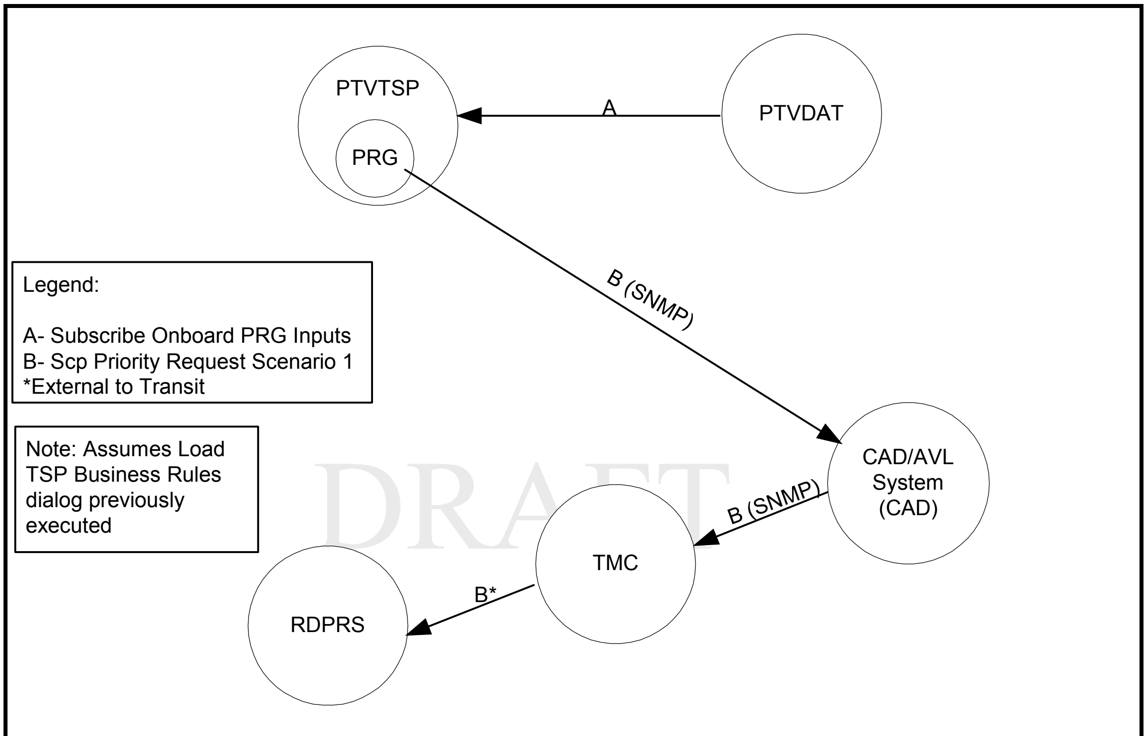


Figure 5.3.2.6.1
Scp Scenario 1 Dialog Flows

5.3.2.6.2 Priority Request Scenario 2

This request scenario is specified by NTCIP 1211 to include the Priority Request Generator in the “Fleet Management Center”. This equates to the CAD/AVL System in the TCIP Model Architecture. The priority requests are sent to the Priority Request Server via the Traffic Management Center (TMC). The effect is that this scenario looks like scenario #1 to the TMC.

The CAD/AVL System determines the vehicle is approaching an intersection based on vehicle location determined either via the “Notify Polling Result” dialog or the “Subscribe PTV AVL” dialog, depending on the agency architecture. The CAD/AVL has the business rules necessary to determine the appropriate priority request strategy and to forward that request to the TMC.

The “Scp Priority Request Scenario 2” dialog uses SNP-encoded messages. Figure 5.3.2.6.2 illustrates the dialog flows for scenario #2.

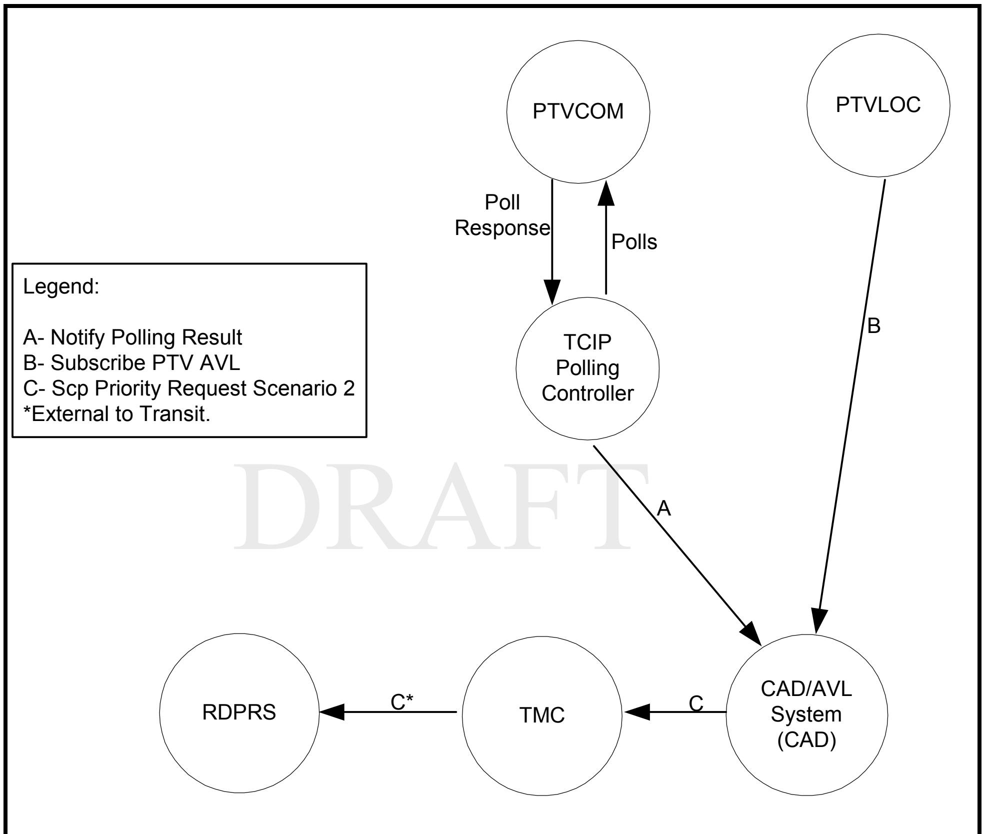


Figure 5.3.2.6.2
Priority Request Scenario 2

5.3.2.6.3 Priority Request Scenario 3

This request scenario is specified by NTICP 1211 to include the Priority Request Generator in the Traffic Management Center. Information to support the creation of a priority request is sent to the TMC by the CAD/AVL System based on location reporting from the PTV. Location reporting may be via the "Notify Polling Result" dialog or the "Subscribe PTV AVL" dialog depending on the agency architecture. Note that the "Subscribe CC PRG" Inputs dialog does not use SNMP encoding. The TMC has business rules to determine the appropriate priority request strategy and to forward that request to the PRS.

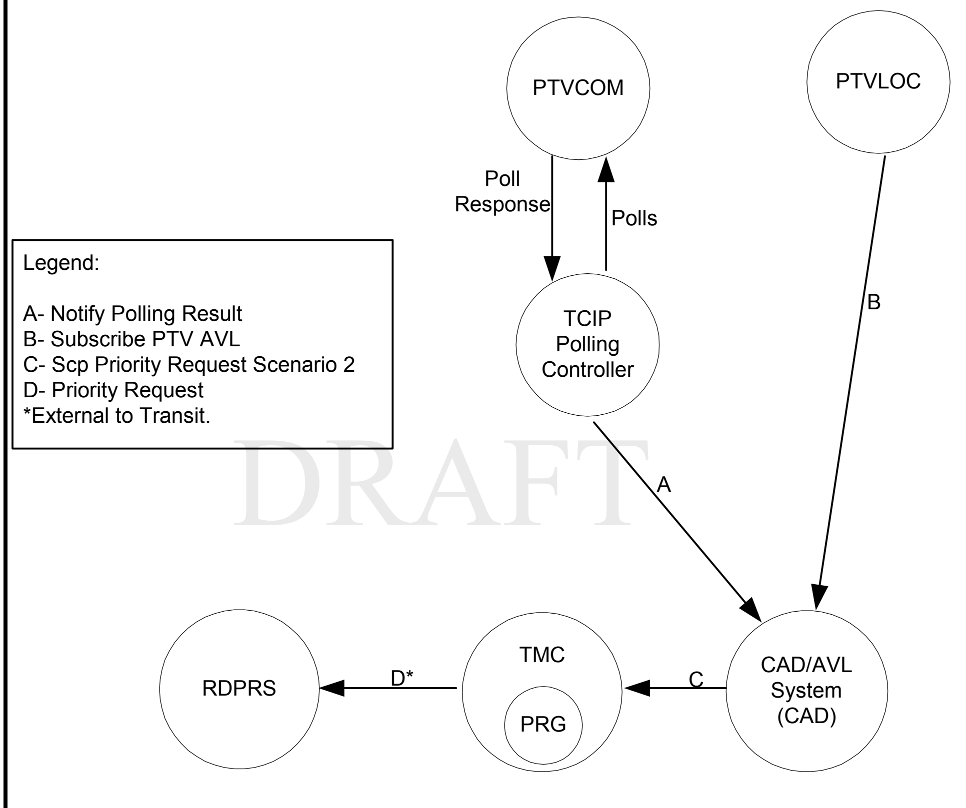


Figure 5.3.2.6.3
Priority Request Scenario 2

5.3.2.6.4 Priority Request Scenario 4

Scenario 4 is implanted with a PTV-based Priority Request Generator (PRG) that sends priority requests to the Priority Request Server (PRS) without going through the CAD/AVL System or the TMC. PTVTSP is loaded with business rules during the preparation for in service operations phase. These business rules allow the PRG within PTVTSP to determine when to request priority. This scenario has two variations: an NTCIP 1211 message-based variation, and an optical variation common in many legacy implementations.

In the optical variation the PRG within PTVTSP requests signal priority by activating an optical emitter via the PTV's VAN.

In the message-based variation, the PTG within PTVTSP determines the appropriate priority request strategy to use and sends the priority request based on the business rules.

Figure 5.3.2.6.4 illustrates the dialog flows for Priority Request Scenario 4.

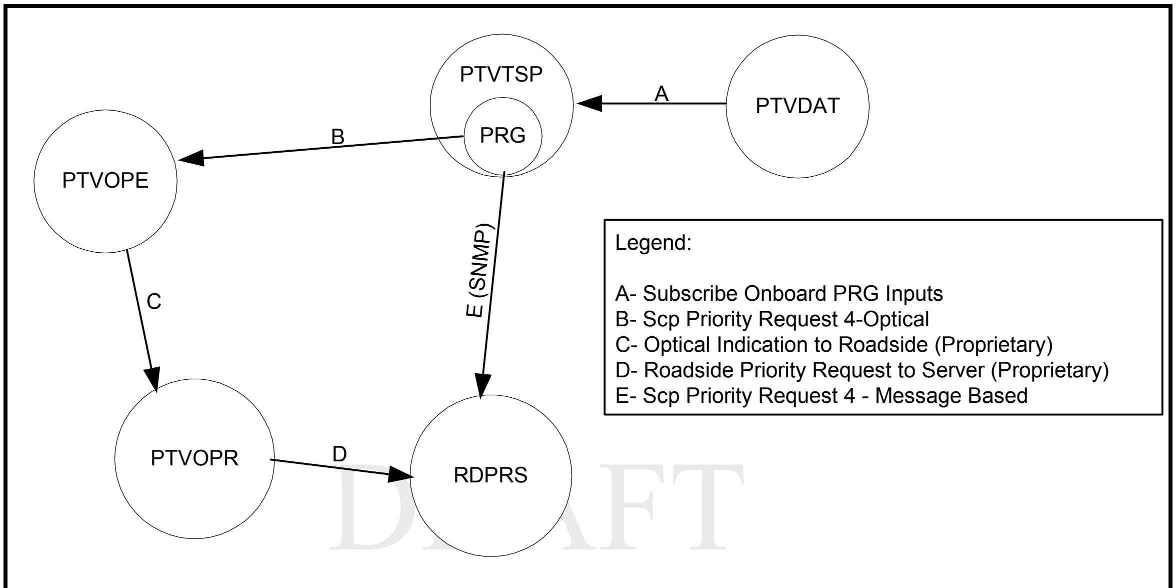


Figure 5.3.2.6.4
Priority Request Scenario 4

5.3.2.6.5 Priority Request Scenario 5

Scenario 5 was added to accommodate implementations where the PTV communicates directly to the roadside, but does not generate priority requests onboard. In this case, PTVTSP sends priority request precursor information to a roadside-based PRG which generates a priority request to the PRS. The roadside PRG contains necessary business rules to generate the request, and PTV requires a load of business rules to allow it to send the precursor information. Figure 5.3.2.6.5 illustrates the dialog flows for Scenario 5.

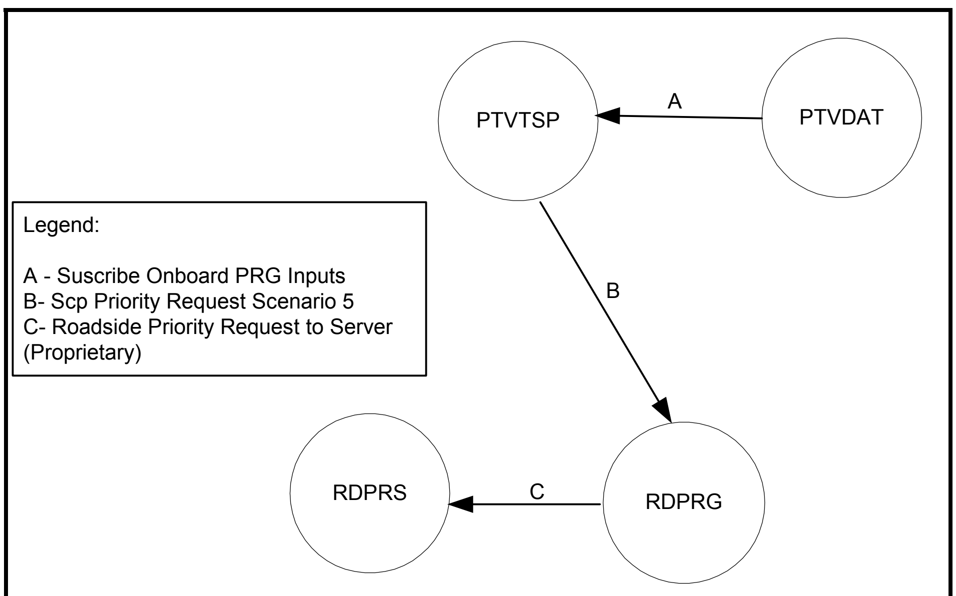


Figure 5.3.2.6.5
Priority Request Scenario 5

5.3.2.7 MDT Menu Sharing

As the PTV operates, onboard devices may need to interact with the PTV operator. These interactions may be simply to display a message to the operator and obtain an acknowledgement, or may require the operator to select a response from a list of alternatives (menu). There are two onboard mechanisms to accomplish this: via the VAN and via TCIP.

VAN-based menuing is defined by SAE J1587.

TCIP-based messaging is based on the Report Menu Selection dialog. Figure 5.3.2.7 illustrates the dialog flows for these interactions.

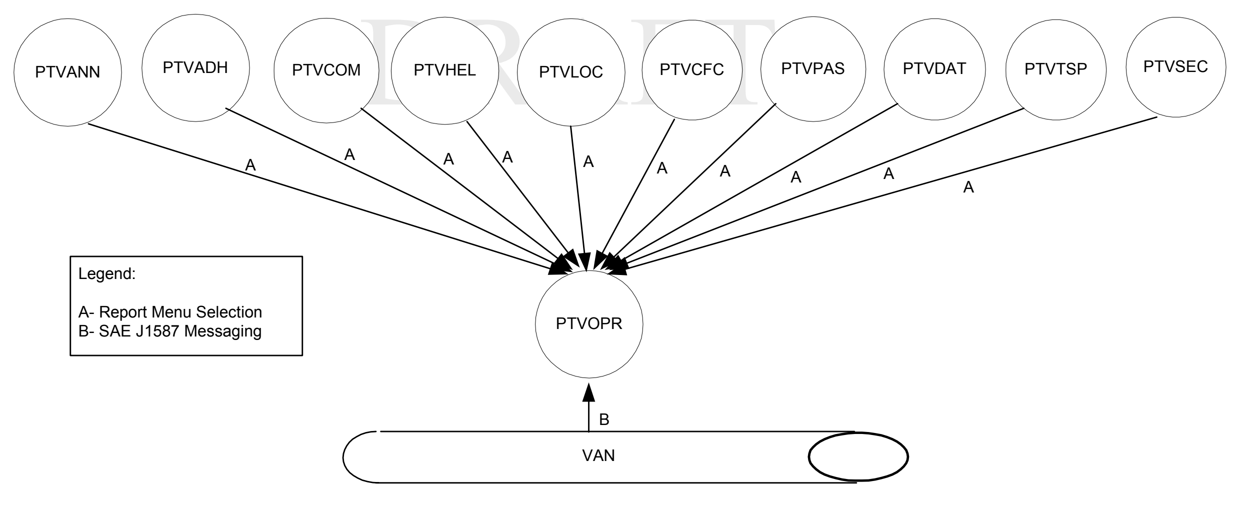


Figure 5.3.2.7
MDT Menu Sharing

5.3.2.8 Passenger Counting

Some PTVs are equipped with automatic passenger counting equipment. Passenger counts are useful for reporting trips provided, for determining TSP strategies, and for validating fares collected. The logical entity in the TCIP model architecture associated with this function is PTVPAS. This entity may be packaged in the VLU, farebox or other onboard component. Figure 5.3.2.8 illustrates the dialog flows for passenger counting.

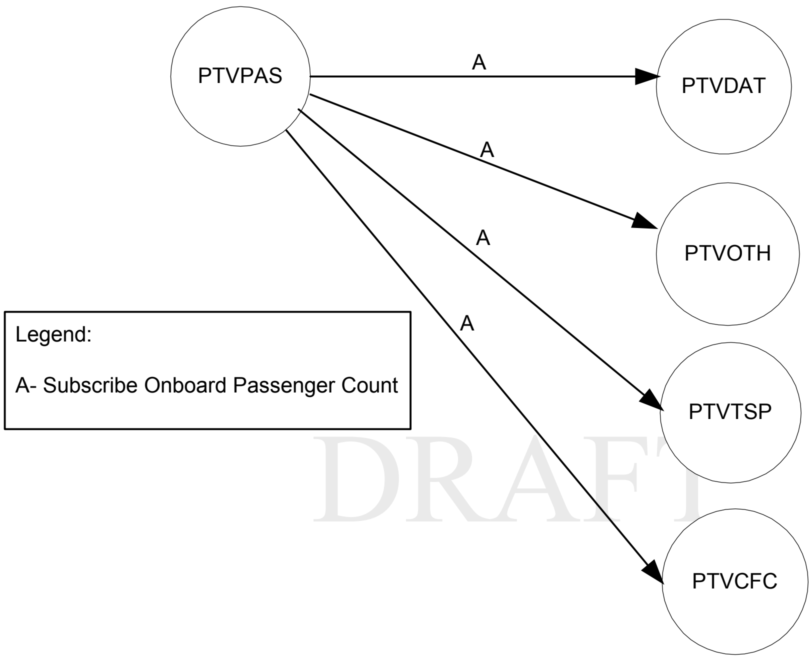


Figure 5.3.2.8
Passenger Counting Dialog Flows

5.3.3 Exceptions to Normal Operations

Section 5.2.4 discusses incidents including dialog flows associated with incident reporting and management. That discussion includes detours and alarm management (silent, operator, and passenger) and is not replicated here. This section handles schedule and route adherence exceptions and enroute trip changes.

5.3.3.1 Route Adherence

The route adherence function monitors and notifies the dispatcher whether a PTV is deviating from its assigned path for its current trip. Route adherence monitoring may be performed onboard the PTV by the PTVADH entity, or centrally by the CAD/AVL System. The CAD/AVL System can perform this function by comparing vehicle location (obtained from the "Notify Polling Result" dialog or the "Subscribe PTV AVL" dialog), with the expected vehicle path based on the schedule for the assigned trip.

If the route adherence function is performed onboard, adherence information is provided to the CAD/AVL System either via PTV poll responses and the "Notify Polling Result" dialog or the "Subscribe PTV Adherence" dialog. Default parameters associated with onboard adherence monitoring are provided to the PTV by the "Load PTV Alarm Limits" dialog during the preparation for in-service operations phase.

The frequency of off-route reporting using the "Subscribe PTV Adherence" dialog is defined by the "Load PTV Alarm Limits" dialog, however, the dispatcher can instruct the CAD/AVL System to send a revised query to

PTVADH to adjust these parameters, (off route distance, and reporting rate) or to cancel the adherence subscription. Note that canceling the adherence subscription impacts schedule adherence reporting as well as route adherence reporting. The dispatcher can effectively disable route adherence only by setting the minimum off route distance to a large value.

The frequency of off-route reporting using polling is governed by the poll cycle length on the radio channel. The dispatcher may, however, use the CAD/AVL System to put selected PTV(s) in a fast poll mode due to their off-route status. This is accomplished using the "Subscribe PTV-Polled Parameters" dialog.

The parameters provided in the "Load PTV Alarm Limits" dialog provide ability to specify criteria under which a vehicle is determined to recover from off-route status. These include: transiting a specified number of timepoints or stoppoints along the original route, or returning to within a specified distance of the original route.

Figure 5.3.3.1 illustrates route adherence dialog flows.

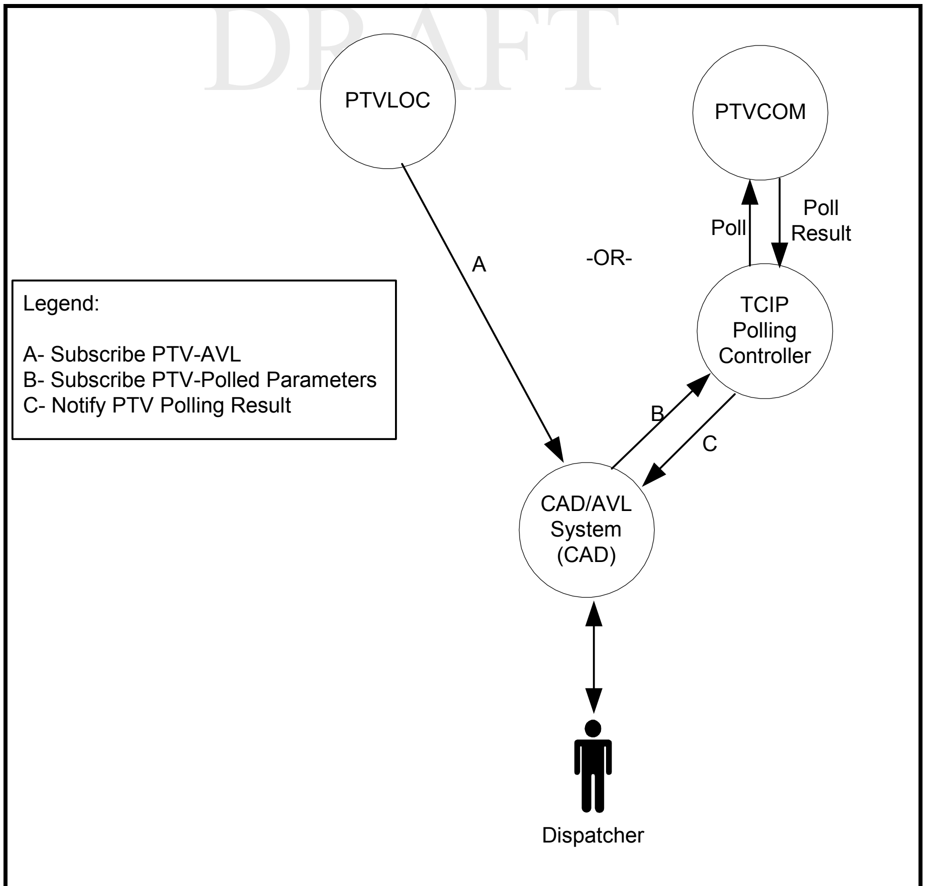


Figure 5.3.3.1.A
CAD/AVL Based Route Adherence Monitoring

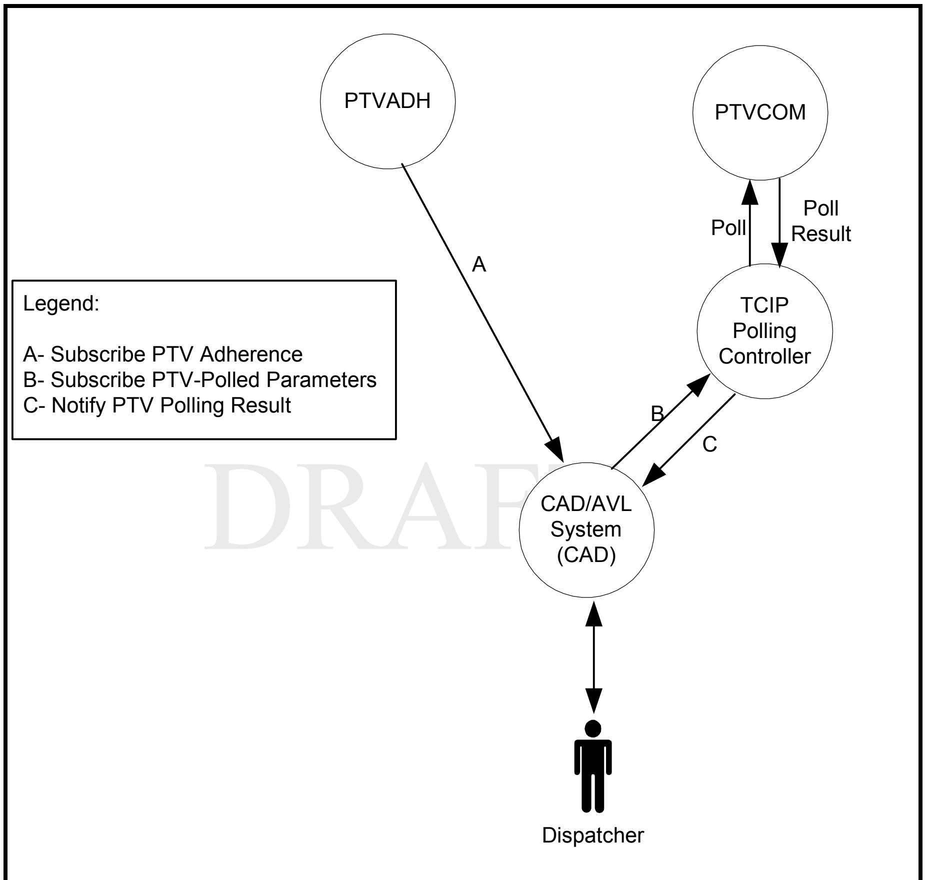


Figure 5.3.3.1.B
CAD/AVL Based Route Adherence Monitoring

5.3.3.2 Schedule Adherence

The route adherence function monitors and notifies the dispatcher whether a PTV is maintaining its assigned schedule for a current trip. Usually schedule adherence is measured by comparing actual and scheduled times at timepoints included in the trip pattern. Schedule adherence monitoring may be performed onboard the PTV by the PTVADH entity, or centrally by the CAD/AVL System. The CAD/AVL System can perform this function by comparing vehicle location send times (obtained from the "Notify Polling Result" dialog or the "Subscribe PTV AVL" dialog), with the scheduled timepoints for the assigned trip.

If the schedule adherence function is performed onboard, adherence information is provided to the CAD/AVL System either via PTV poll responses and the "Notify Polling Result" dialog or the "Subscribe PTV Adherence" dialog. Default parameters associated with onboard adherence monitoring are provided to the PTV by the"Load PTV Alarm Limits" dialog during the preparation for in-service operations phase.

The schedule adherence criteria using the Subscribe PTV Adherence dialog is defined by the "Load PTV Alarm Limits" dialog, however, the dispatcher can instruct the CAD/AVL System to send a revised query to PTVADH to adjust these parameters, (off schedule early late tolerance) or to cancel the adherence subscription. Note that canceling the subscriptions impacts route adherence reporting as well as schedule adherence reporting. The dispatcher can effectively disable schedule adherence by setting the off schedule tolerance to large values.

The location reporting frequency using polling is governed by the poll cycle length on the radio channel. The dispatcher may, however, use the CAD/AVL System to put selected PTV(s) in a fast poll mode due to their off-route status. This is accomplished using the "Subscribe PTV-Polled Parameters" dialog.

Figure 5.3.3.2 illustrates schedule adherence dialog flows.

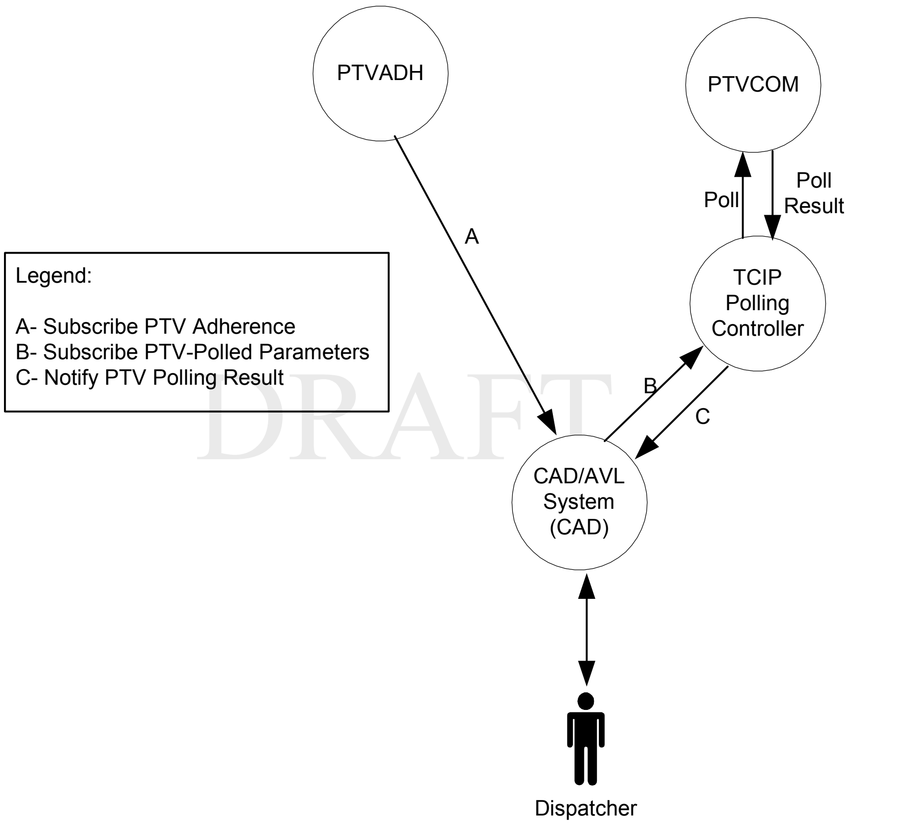


Figure 5.3.3.2
CAD/AVL Based Schedule Adherence Monitoring

5.3.3.3 Enroute Trip Changes

Occasionally it becomes necessary to change plans for a PTV after it has departed the garage. Section 5.3.4 discussed the ability for the dispatcher to implement a detour. In some cases it is not appropriate to use a detour, and instead it is necessary to change a PTV's assignment to another scheduled trip or to a newly created trip or set of trips.

If the new trip(s) have already been defined to PTVDAT using the "Load Schedule" dialog in the preparation for in-service operations phase, the dispatcher can simply reassign the PTV from one trip to another. The "Command Change Assignments" dialog accomplishes this function. Agency-specific policies and procedures govern the process for moving between assigned trips including discharging passengers, going out of service, moving to the appropriate location to begin the newly assigned trip etc. Normally, this will include a voice conversation between the dispatcher and the PTV operator.

If the new trip has not already been defined to PTVDAT, then it is necessary to define the trip along with the reassignment. This is accomplished using the "Command Load PTV Trips" dialog. Agencies must use caution in implementing this dialog, as the message to the PTV converting the new trip(s) host the potential to be large and impact the radio link performance, or cause the message to be rejected by the TCIP Polling Controller.

Figure 5.3.3.3 illustrates the dialog flows for enroute trip changes.

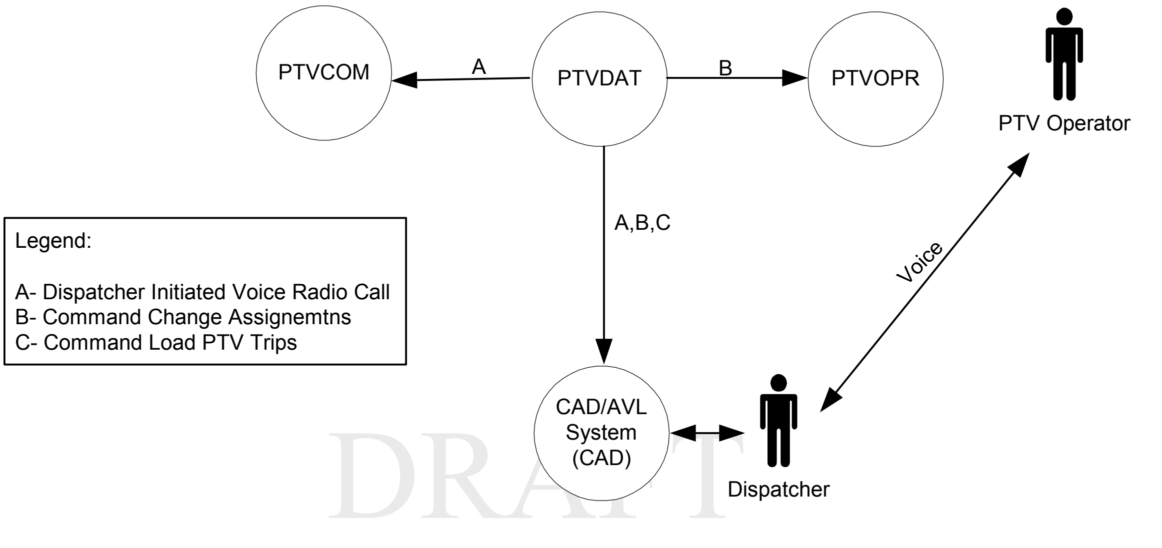


Figure 5.3.3.3
Enroute Trip Changes

5.3.4 Close Out of Normal Operations

After completing service the PTV normally returns to the bus garage. A variety of post service activities may take place including:

- Recording the pull-in to the garage
- Removing cash and tokens from the farebox
- Unloading data from the PTV
- Logoff and shutdown

In addition to the individual PTV closeout operations, there is a closeout with the Traffic Management Center, and another closeout with PTSF entities. These closeouts may occur on a daily or other periodic basis specified by local policies.

Figure 5.3.4 illustrates the dialog flows for the closeout of normal operations.

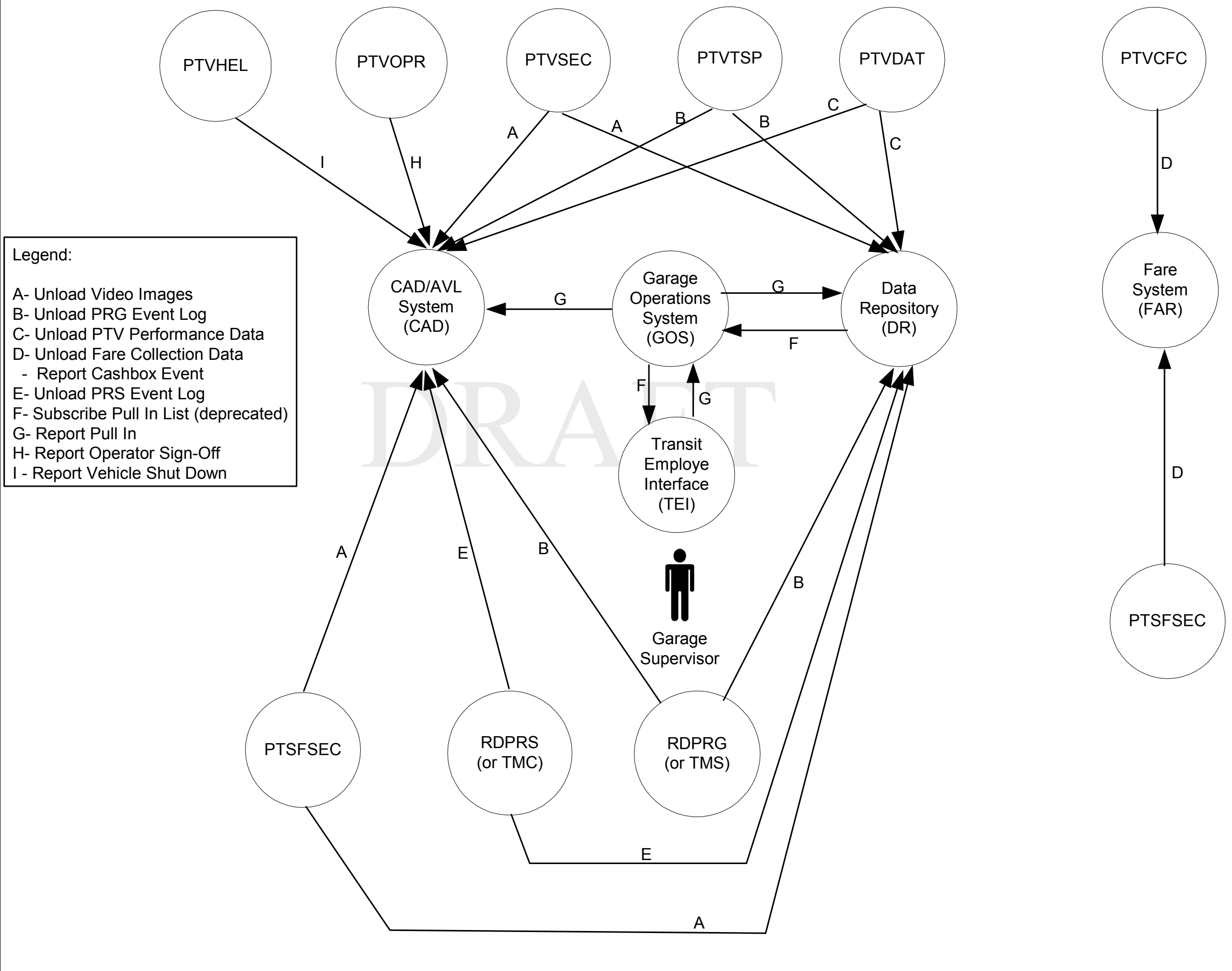


Figure 5.3.4
Closeout of Normal Operations Dialog Flows

5.3.4.1 Recording the Pull-In

The PTV's final trip of the day will normally be an out of service trip containing a final timepoint at the garage. The scheduled timepoint time for this final timepoint and the scheduled pull-in time will correspond.

The garage-supervisor will obtain a list of scheduled pull-ins from the Garage Operations System (GOS), using a Transit Employee Interface (TEI). The supervisor will report the pull in, along with the actual pull-in time. Pull-In times may be significant to an agency due to overtime liability, labor agreements, etc.

5.3.4.2 Removing Cash and Tokens

The cashbox and tokens may be removed at the garage, or the bus may go to a separate location to have the cashbox removed. Cashboxes may be removed nightly, or allowed to remain on the PTV for a period of days depending on local agency policies. When the cashbox is removed by an authorized employee, the even and the cashbox contents are reported by the PTV to the Fare System.

5.3.4.3 Unloading Data from the PTV

The logical entities onboard the PTV collect and log data throughout the work assignments. When the PTV returns to wireless LAN coverage, this information needs to be removed (unloaded) from the PTV to a longer term data store. Local agency architectures define what specific store(s) to use for what data unloads in a specific agency.

Data to be unloaded includes:

- Fare Collection Data
- PTV Performance Data which includes vehicle time/motion information as well as vehicle health and alarm information
- Transit Signal Priority Data (PRG Event Log)
- Video Images captured onboard the PTV

5.3.4.4 Logoff and Shutdown

The final event of a work day from a PTV perspective, are the sign off by the PTV Operator and possibly a PTV Shutdown. Note that some agencies to not shutdown PTVs after a work block-especially in colder climates. Also, many implementations do not shutdown the computers onboard a PTV promptly when the engine is shut off. This may allow data unloads to complete prior to computers shutting down and effectively disabling the various PTV logical entities.

5.3.4.5 Non PTV Closeouts

The periodic closeout with the traffic management area obtains TSP history data from TMC and field based PRG(s) and/or PRS(s). This provides a bases for evaluating the effectiveness of the TSP System(s) installed.

The periodic closeout with the stoppoint of station entities includes cashbox event reporting, fare collection data unloading and video image recovery from PTSFCFC and PTSFSEC.

5.4 Revenue Management and Fare Collection

The Revenue Management and Fare Collection business process deals with all issues related to setting fare policy, determining fares, collecting fares, and dealing with exceptions encountered in the fare collection process.

Fare policies are locally determined based on a complex process involving local and regional political considerations, agency revenue needs and sources, ridership forecasts and other factors. Since fare policies are highly agency dependent, TCIP provides some basic capabilities which may be tailored to individual agency needs.

The transit industry is currently developing a Universal Transit Farecard Standard (UTFS), sometimes called a smartcard. Many aspects of the Revenue Management and Fare Collection business process will fall under the UTFS standardization umbrella including back office communications with financial institutions, farecard content, validation security requirements, and reader/writer requirements.

The TCIP Fare Collection Business area deals with a subset of the overall business process which includes:

- Monitoring Farebox Health
- Reporting Cashbox Events (e.g. removal)
- Loading and Unloading Fare Collection Data to Collection Equipment in a PTSF or PTV
- Reporting Farebox data load errors
- Calculating fare for a proposed transit itinerary

This process has 3 stages:

- Preparation for revenue service
- Revenue Service
- End of Revenue Service

In preparation for Revenue Service PTVCFC or PTSF verifies that it has the correct software, farebox configuration data, and fare data (e.g. farezones). If the latest data is not available it is loaded from the Data Repository or Fare

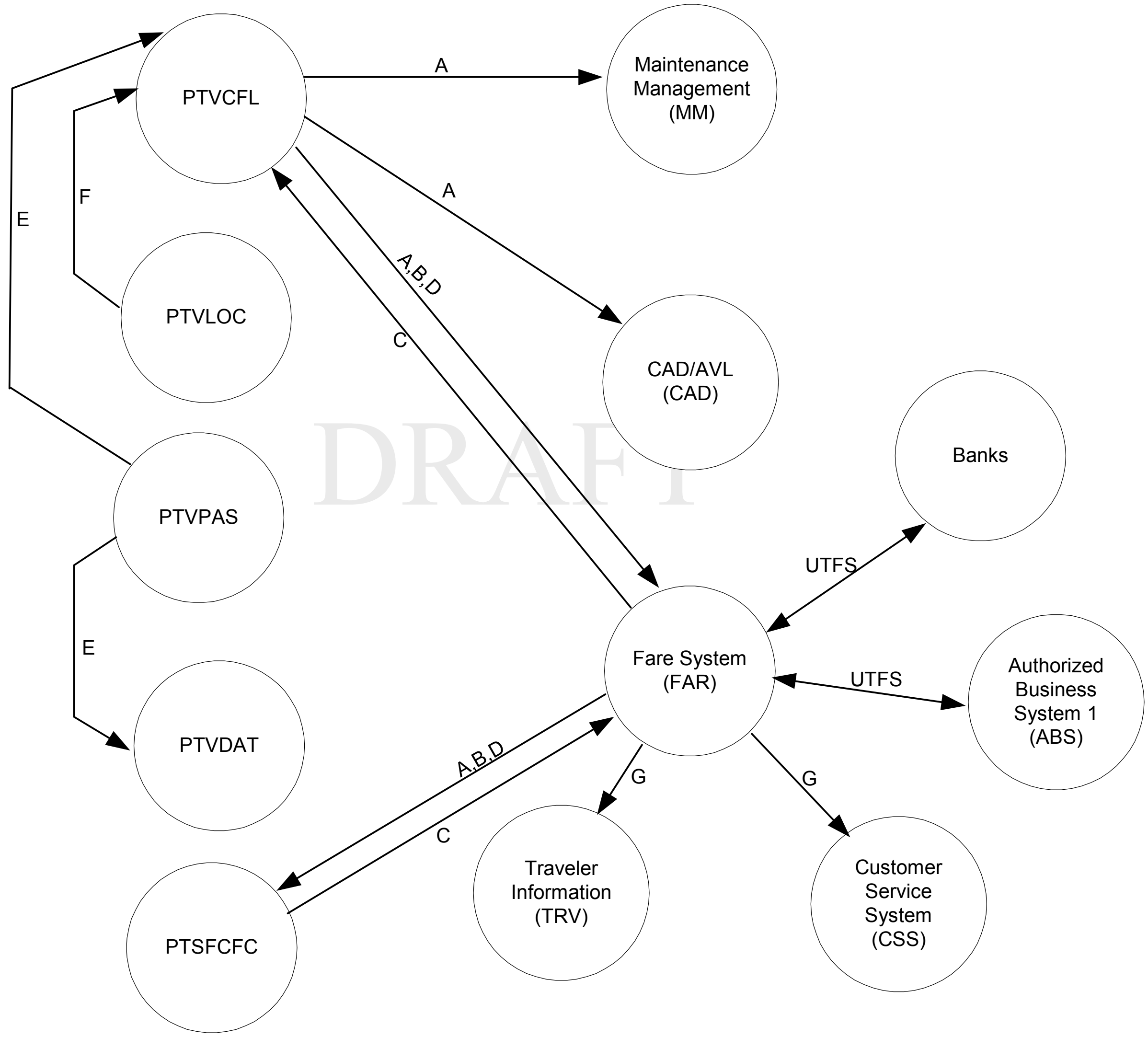
System. Note: in the case of a vehicle-borne farebox (PTVCFC), the data load is only performed when in wireless LAN coverage. If the loaded data is not valid, this is reported to the fare system.

During revenue service, PTVCFC or PTSF manages fare collection. Any health impairment or unexpected event is reported to the CAD/AVL System, Maintenance Management System and Fare System. If the farebox is removed or tampered, this is reported to the Fare System.

At the end of revenue service for PTVCFC, or at an agency designated time or even for PTSFCFC, the entity unloads its daily operating data to the Data Repository or the Fare System. Cashbox removal and installation events are reported to the fare system, including the calculated cashbox contents.

At any stage in the process, the Customer Service System or the Traveler Information System may request the Fare System to calculate a fare for a specified itinerary to assist customers in planning future transit trips.

Figure 5.4 shows the TCIP dialog flows for Revenue Management and Fare Collection. Note that the onboard fare collection entity receives passenger count information from the onboard passenger counter entity, however the farebox may contain both entities. If the farebox contains both entities, then the farebox reports passenger counts to the onboard data manager entity (PTVDAT).



Legend:

A- Subscribe Fare Collection Health
B- Report Cashbox Event
Report Farebox Validation Error
C- Load Component Configuration Data
Load Component Software
Load Fare Collection Data
D- Unload Fare Collection Data
E- Subscribe Onboard Passenger Count
F- Subscribe Onboard Location
G- Subscribe Itinerary Fare
H- Subscribe Daily Revenue Data

Figure 5.4.A
TCIP Component of Revenue Management and Fare Collection Function Without Data Repository

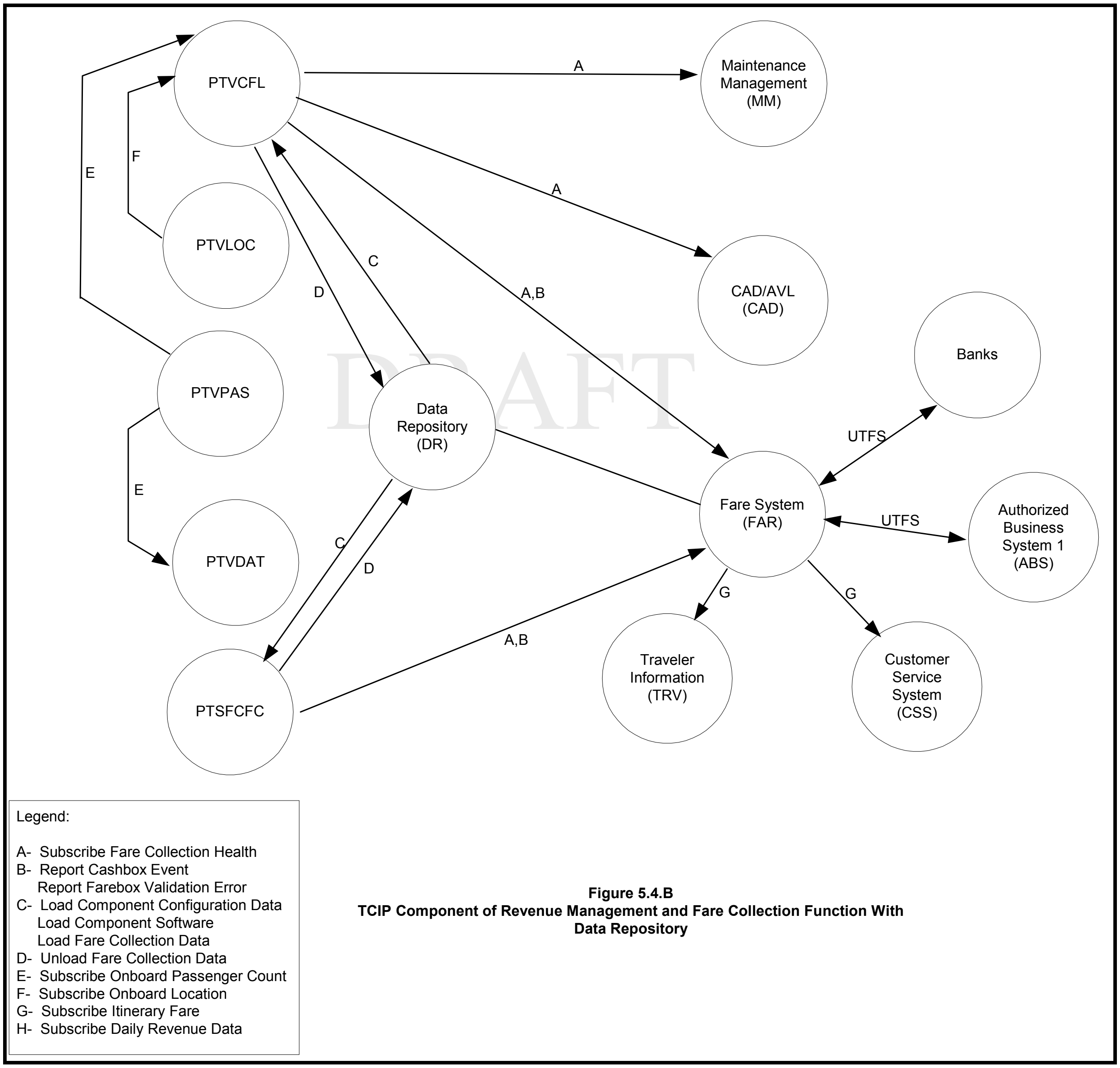


Figure 5.4.B
TCIP Component of Revenue Management and Fare Collection Function With Data Repository

5.5 Schedule Creation and Distribution

This business process creates transit schedules and distributes those schedules to a data repository or to other business systems that need them.

5.5.1 Scheduling Processes

The schedule development process in transit includes three key processes to define and manage transit service: schedule writing, block building and run-cutting.

5.5.1.1 Schedule Writing

Schedule writing is the process of creating a route and defining the service that will operate that route. A route is defined by one or more patterns – the geographic paths over which trips travel. Patterns may contain many types of points and events, including timepoints, bus stops, transfer points, fare zone changes, destination sign changes, transit signal priority triggers, operator road relief points, automated announcements to passengers and other messages to the operator.

Timepoints are exact locations along routes where trips are assigned specific arrival and departure times. Running time is defined for any timepoint pair and may be used as a system-wide default for all routes traveling between the pair. Some agencies maintain a separate running time table for each route, in order to more reflect the different operating conditions affecting various routes. Running times may also vary by time of day , day or week, based on weather, etc. Trips are built from a time at a particular timepoint in a particular pattern using default running times for the remaining timepoint pairs in the pattern.

Figure 5.5.1.1.A depicts the process of gathering source data into the Scheduling System to use in the schedule writing process. The Scheduling/Blocking/Runcutting System may edit/add/or delete timepoints, pattern segments, and patterns to create new and/or revised routes as part of a new schedule. It may also use past operating data such as passenger loading by time of day to forecast service demand as an aid to planning and scheduling service frequencies. The source of each type of information is dependent on the agency's local architecture.

A TCIP transit schedule is distributed as a linked series of artifacts. The "Subscribe Master Schedule Version" dialog provides information on the version number(s) in effect for these artifacts for specified route(s) for an interval of time. Several versions of an artifact can be available for use at a point in time since agencies generally have need for a variety of schedules available for use. Such as weekday, holiday, Saturday, Sunday, and special event. The artifacts for which the "Subscribe Master Schedule Version" provides version numbers are:

- Route Schedule
- Patterns
- Timepoints
- Stoppoints

Timepoints and stoppoints are geographical locations which are included in patterns. Patterns specify a series of transit vehicle movements. A route is a named (or numbered) series of patterns over which service is provided to the public (e.g. the #9 Bus Route, or the Blue Route). Revenue trips are the specific instances of a transit vehicle traversing a route to provide service. Other types or trips are defined for administrative purposes such as road test, garage transfer etc. Notes provide textual information about any of the other artifacts, and are referenced in those artifacts by a note ID number.

Timepoints, stoppoints, and patterns, are transferred in TCIP using the "Subscribe Time Point List", "Subscribe Stop Point List", and "Subscribe Pattern List" dialogs. Since the same timepoints, stoppoints, and pattern version(s) may be applicable to multiple schedules separate distribution of these artifacts reduces the need to resend identical information when distributing schedules.

Routes are not distributed as a unique information flow. The reason is that many agencies vary the patterns within a route from schedule to schedule, or from trip to trip within a daily schedule. For example, the bus that follows an arterial past the entrance to a community college may vary its pattern during certain times on weekdays to service a stop point just inside the college. The schedule for a route is transferred using the "Subscribe Route Schedule" dialog.

The Route Schedule message includes the following key information about a route and the associated schedule:

- <u>Time Interval</u> for which the message is applicable
- <u>Default Patterns</u> for each direction associated with the route
- <u>Scheduled Trips</u> for each direction for the route for the specified time interval
- <u>Trip Specific Patterns</u> associated with scheduled trips which deviate from the default patterns for that route.

Figure 5.5.1.1.B provides a conceptual overview of the relationships of schedule artifacts. A higher block resting on a lower block implies that the higher block is dependent on the information in the lower block. For instance, Patterns rest on stoppoints and timepoints because a pattern is defined as a series of stoppoints and/or timepoints.

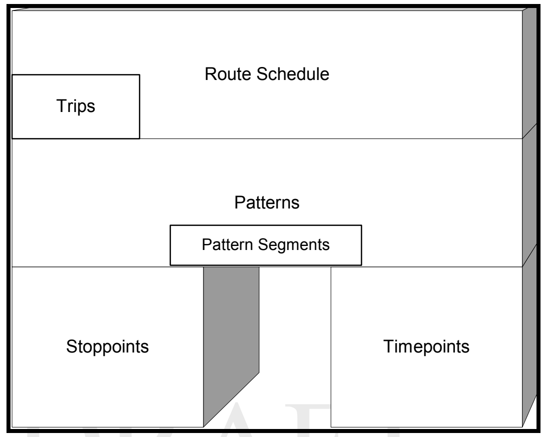


Figure 5.5.1.1.B
Structure of Schedule Definitions in TCIP

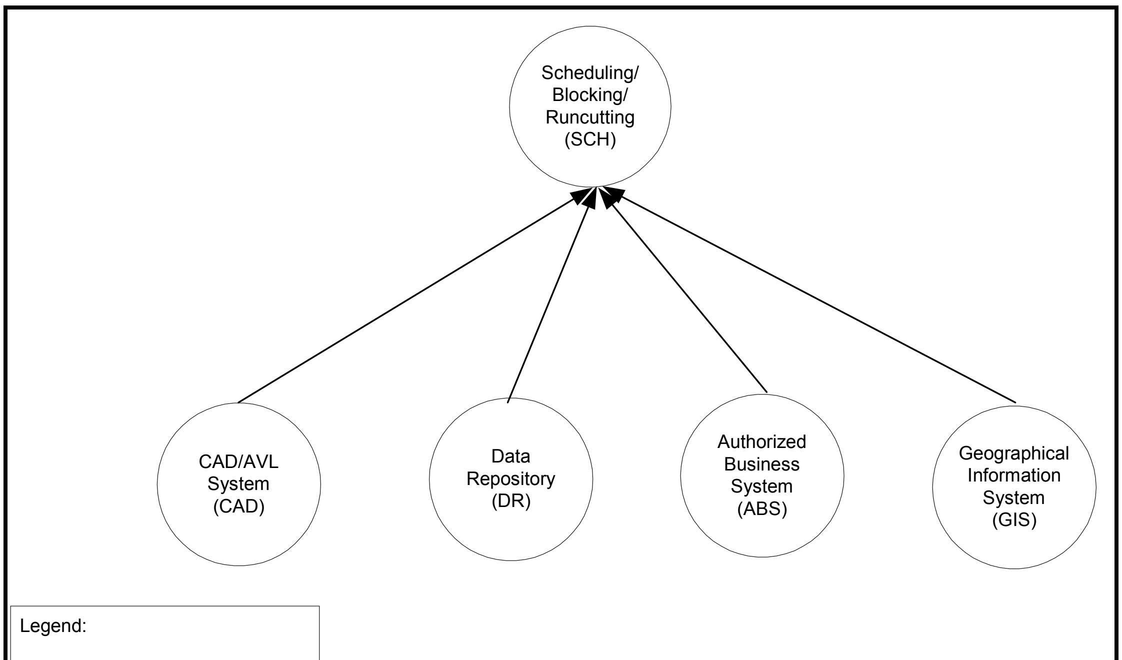


Figure 5.5.1.1.A
Data Gathering Dialog Flows for Schedule Writing

5.5.1.2 Block Building

Once schedules are written, the block building process combines trips into optimized vehicle assignments such that the number of PTVs and platform hours are minimized. A block, also known as a vehicle assignment, includes everything a PTV is assigned to do from the time it pulls out of the garage until it pulls in. Different vehicle types are assigned to specific trips or routes based on ridership requirements or other characteristics of the route. The block building process determines the amount of layover or recovery time that a PTV will have between scheduled revenue trips. This is also the process that identifies the deadhead trips that might be needed to move a PTV from one route to another so that all trips are operated efficiently.

5.5.1.3 Run-Cutting

Run-cutting is the final step in the scheduling process, in which vehicle assignments are cut into optimized operator assignments. A short vehicle assignment may require just one operator, while longer vehicle assignments may require several operators throughout the day. Some pieces of work may be combined to create a split shift assignment for a guaranteed 8-hour day. The goal of the run-cutting process is to efficiently distribute the work so that overall costs are minimized given union contract rules, pay rates, work rules and management requirements.

5.5.2 Scheduling Products

The outputs from the Scheduling/Blocking/Runcutting business processes are"

Route Schedule – Defines the scheduled trips on a bus route in both directions. Trips are defined in terms of the patterns they are to follow and the times that the PTV is scheduled to arrive at (or depart from) timepoints within the pattern.

Patterns- Patterns define the movements of PTVs along a route. A pattern is defined as a sequence of pattern segments. Pattern segments in turn define the timepoints, stoppoints and way points associated with the PTV's movement. A pattern may also define other event that are schedule enroute (e.g. destination sign changes). Patterns and pattern segments can be shared by more than one trip.

Timepoints – Timepoints define locations at which the PTV's progress is measured against schedule. Each trip in the route schedule, is defined not only by the pattern it follows, but by the scheduled times at timepoints for the PTV. Timepoints may be shared by more than one pattern or pattern segment.

Transfers – Transfers define opportunities for transit passengers to switch from one PTV and trip to another – usually between PTVs operating on different routes. The transfers specified in this out put from the scheduling process generally do not include all available transfers, but are limited to a subset that the agency determines to be of particular interest. Transfers may be protected or unprotected. A protected transfer implies that the agency may hold the departing PTV to wait for the arriving PTV so that transferring passengers can make the connection. Agency policies limit the duration and circumstances of such holds. Transfers may also be intermodal (e.g. light rail or subway to bus or vice versa.)

Operator Assignments (unbound) – Operator assignments (or "runs") define daily work assignments for PTV operators. A this stage in the process individual operators have not been paired (bound) with the work assignments, consequently the assignments are termed "unbound". A run generally consist of a series of assigned trips, however some agencies end/begin runs mid-trip. TCIP accommodates this requirement by allowing a run to begin or end or switch to a different trip at any timepoint on any trip.

Vehicle Assignments (unbound) – Vehicle assignments (or "blocks") define a daily work assignment for a PTV. At this stage in the process, specific PTVs have not been paired (bound) with the work assignments, consequently the assignments are termed "unbound".

Running Times - Running times define the scheduled time for a PTV to move from one timepoint to another. These may vary based on time of day , day of week, weather and other factors. Other business systems may require this information to predict arrival times, or for other purposes.

5.5.3 Schedule Distribution and Validation

Once the scheduling products are created, they must be distributed, to other business systems that require them. Those other business systems may:

- Create derivative products as described in other business processes (e.g. rosters, timetables)
- Validate the schedule and determine that it is defective in some way.
- Make changes to the version of the schedule without updating the version number of the schedule (e.g. to add a stoppoint or timepoint to a pattern segment).
- Redistribute the schedule to other applications – as in the case of a Data Repository that obtains the schedule once from the Scheduling/Blocking/Runcutting Business System, and then provides schedule information to all other subscribers.

5.5.3.1 Schedule Validation

since the Transit Schedule is used by many business systems in the agency for different purposes, there are opportunities for these business systems to determine that the schedule is inadequate for a variety or reasons that would not be detected during the Scheduling/Blocking/Runcutting process. TCIP provides the “Report Schedule Validation Error” dialog to allow business systems to report the error to an agency designated business system such as the Data Repository or Scheduling/Blocking/Runcutting System.

Figure 5.5.3.1 depicts the schedule validation error reporting process.

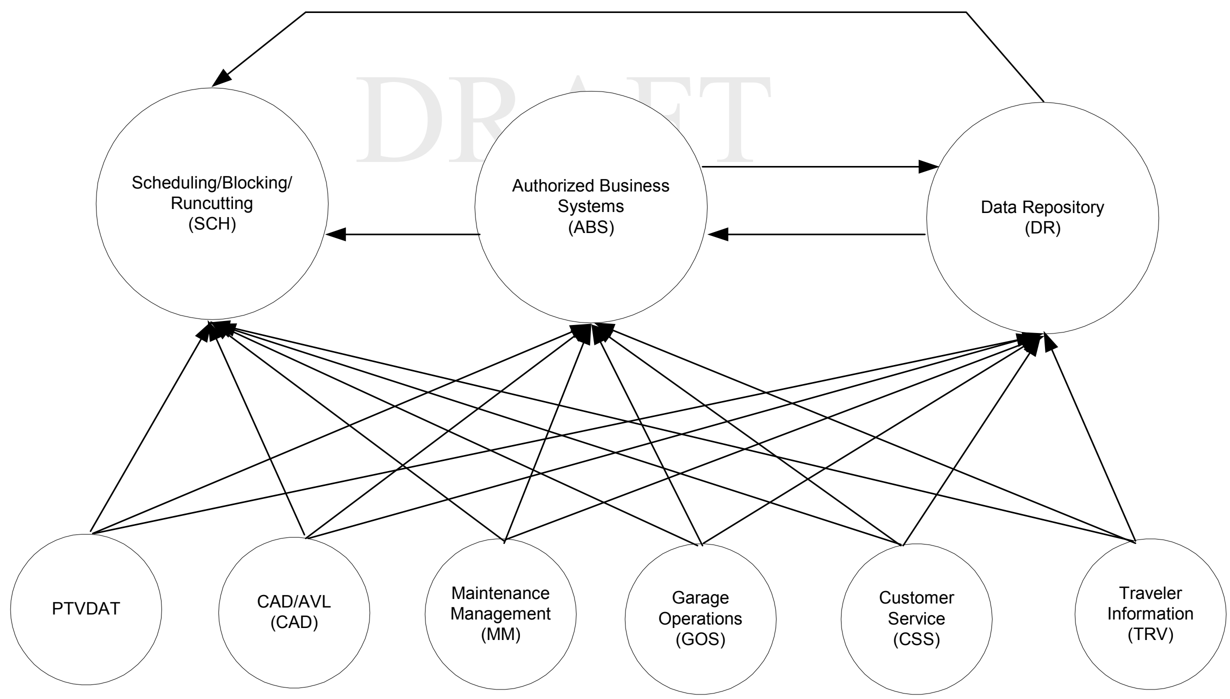


Figure 5.5.3.1
Possible Schedule Validation Dialog Flows

NOTE: All flow lines represent "Report Schedule Validation Error".

5.5.3.2 Schedule Product Distribution

The scheduling products are distributed using the dialogs:

- Subscribe Master Schedule Version
- Subscribe Route Schedule
- Subscribe Pattern List
- Subscribe Timepoint List
- Subscribe Operator Assignment
- Subscribe Vehicle Assignments
- Subscribe Transfers
- Subscribe Running Times

The “Subscribe Master Schedule Version” dialog allows a subscriber to obtain the version number of the RouteSchedule, PatternList, TimepointList and StoppointList (CPT Business Area), for a designated period of time. Note that there may be more than one schedule version in effect in a designated time period (e.g. weekday, weekend, holiday). Knowing the correct version number(s) allows the subscriber to obtain the correct version of each artifact.

Having obtained the artifacts (RouteSchedule, Patterns, Timepoints, and Stoppoints), the subscriber can obtain updates to the version of the artifact using the same dialog that retrieved the artifact originally.
Depending on the agency architecture, the Scheduling products might be transferred from the Scheduling/Blocking/Runcutting system to a Data Repository for further distribution or to individual subscribing applications directly.

Figure 5.5.3.2 depicts the schedule distribution processes.

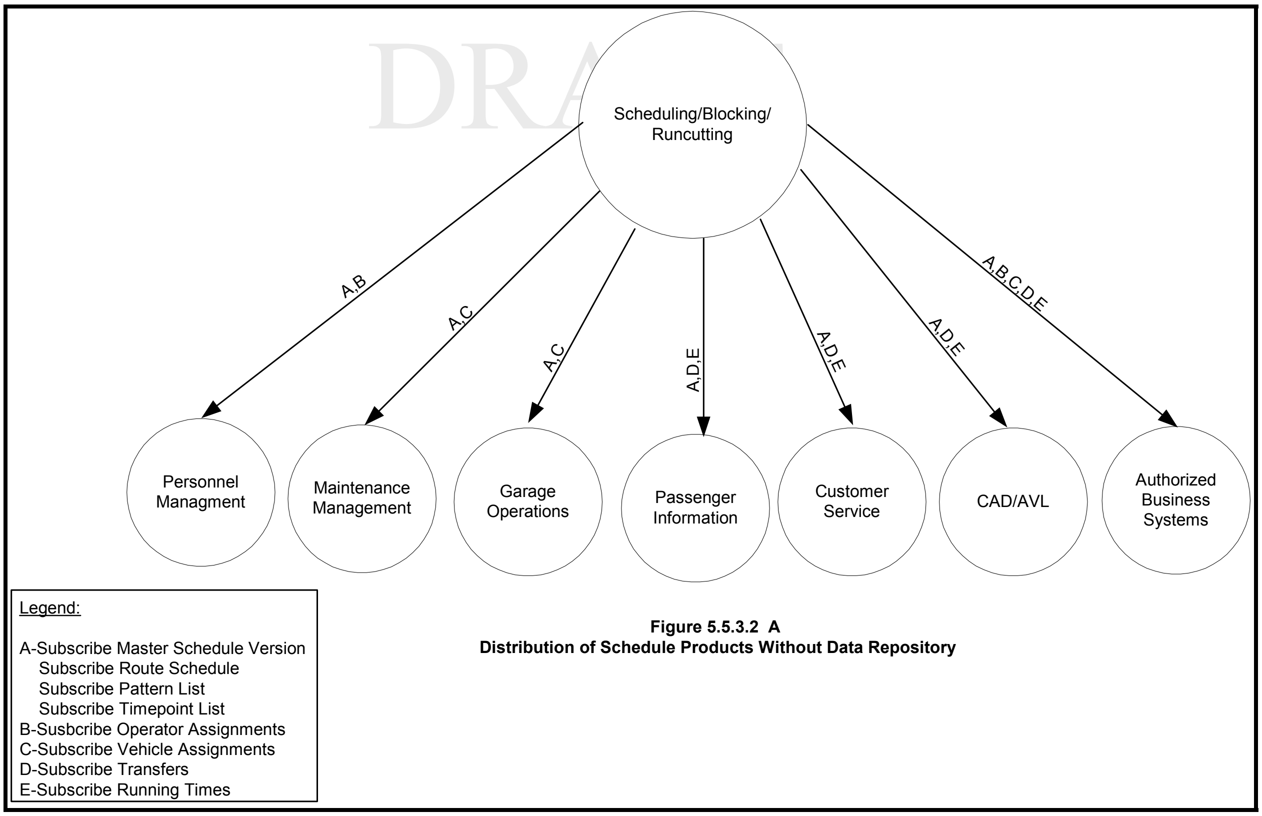


Figure 5.5.3.2 A
Distribution of Schedule Products Without Data Repository

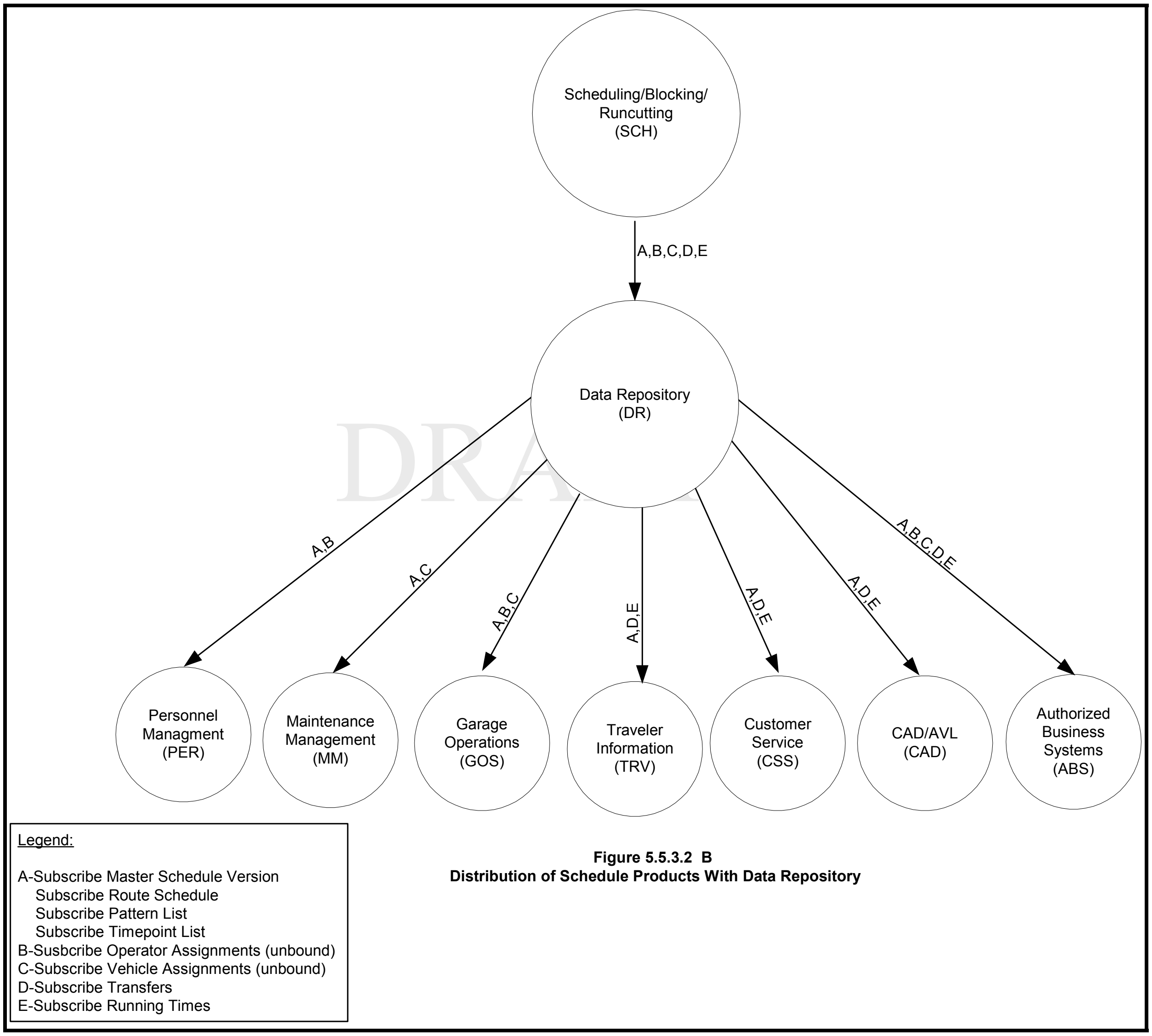


Figure 5.5.3.2 B
Distribution of Schedule Products With Data Repository

5.6 Personnel Management and Assignments

The Personnel Management and assignments business process ensures that service-ready PTV operators, and other employees are available to provide and manage scheduled service, and assigns specific people to work assignments including qualified PTV operators to work assignments (runs) defined by the Scheduling/Blocking/Runcutting System (SCH).

Schedule Operator Assignments (runs) are combined into rosters which define weekly groupings of work assignments. In some agencies, union agreements govern a “pick” process by which PTV operators choose their work on a monthly or quarterly basis. The result s of the pick process are a preliminary set of bindings between scheduled work and specific PTV operators expected to perform that work. The pick process also results in the production of extra lists which indicate what PTV operators are available for duty on each day that do not have work assigned.

The Personnel Management System (PER) obtains the onboard operator assignments for each business day , and combines those assignments into unbound rosters and supports the pick process. Bound Operator Assignments and the extra list are then provided by PER to other business systems.

The bound operator assignments may subsequently be changes as a result of PTV operator illness, unfitness for duty, termination or employment, etc. Figure 5.6 depicts the PTV operator Personnel assignment process.

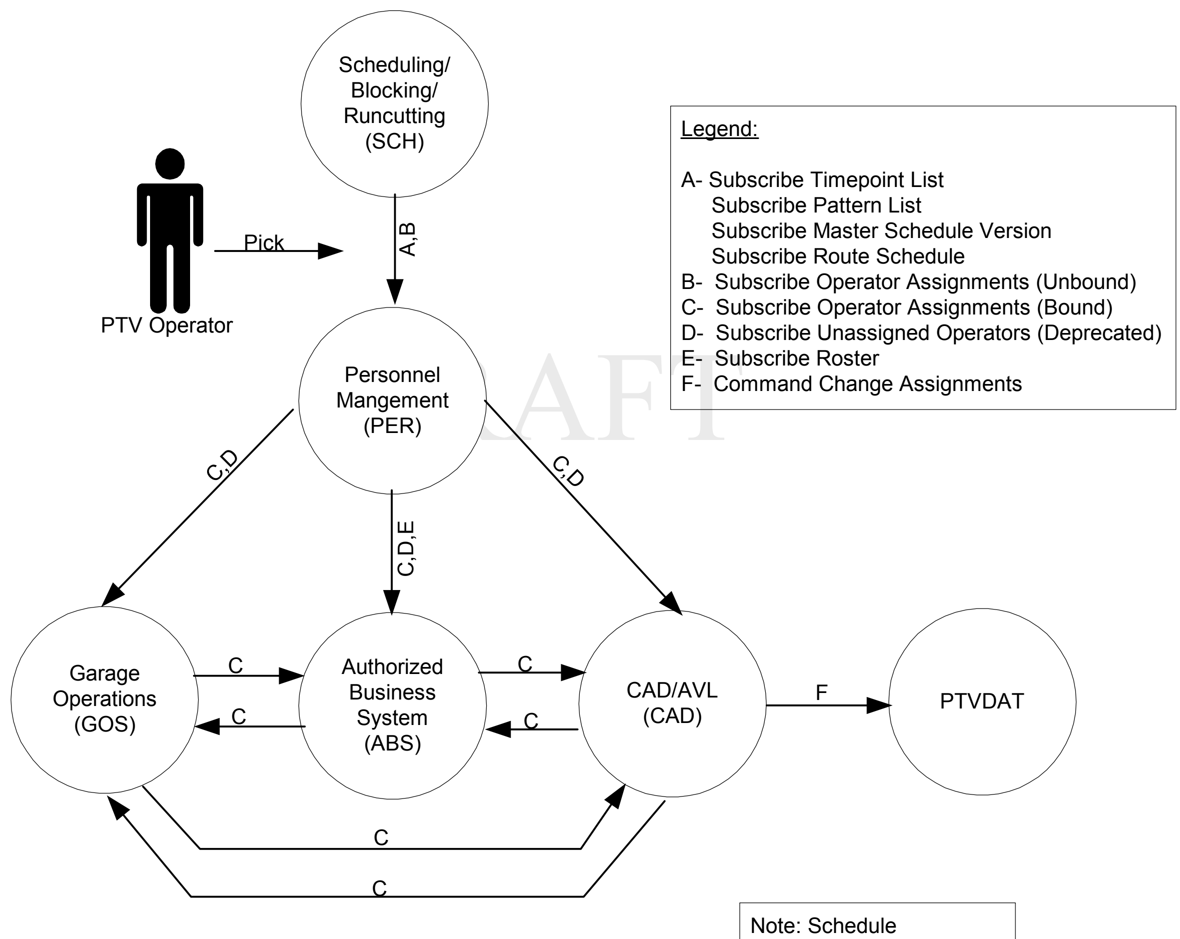


Figure 5.6.1.A
Operator Assignment Distribution Without Data Repository

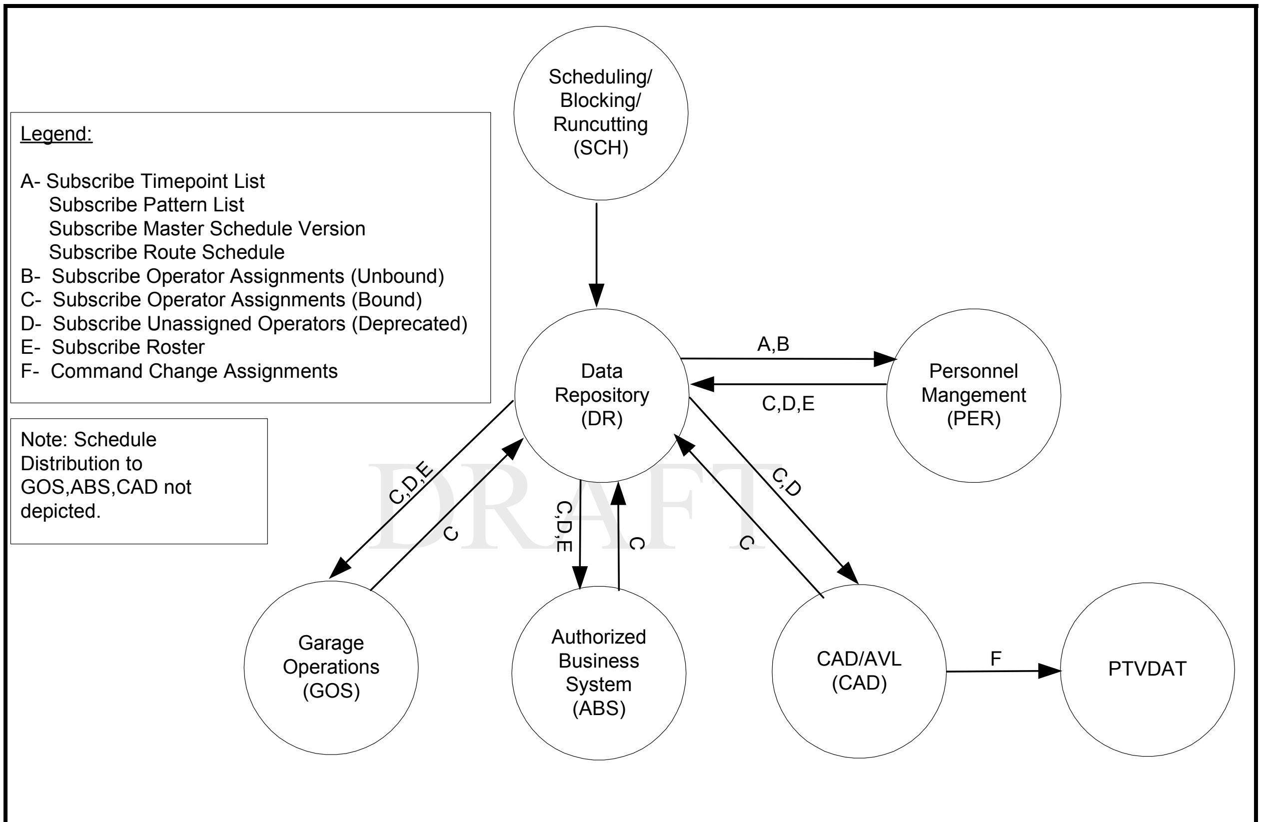


Figure 5.6.1. B
Operator Assignment Distribution With Data Repository

5.7 Vehicle Maintenance & Assignments

The Vehicle Maintenance and Assignments business process ensures that service-ready PTVs are available to provide scheduled service and assigns specific PTVs to work assignments (blocks) defined by the Scheduling/Blocking/Runcutting System (SCH).

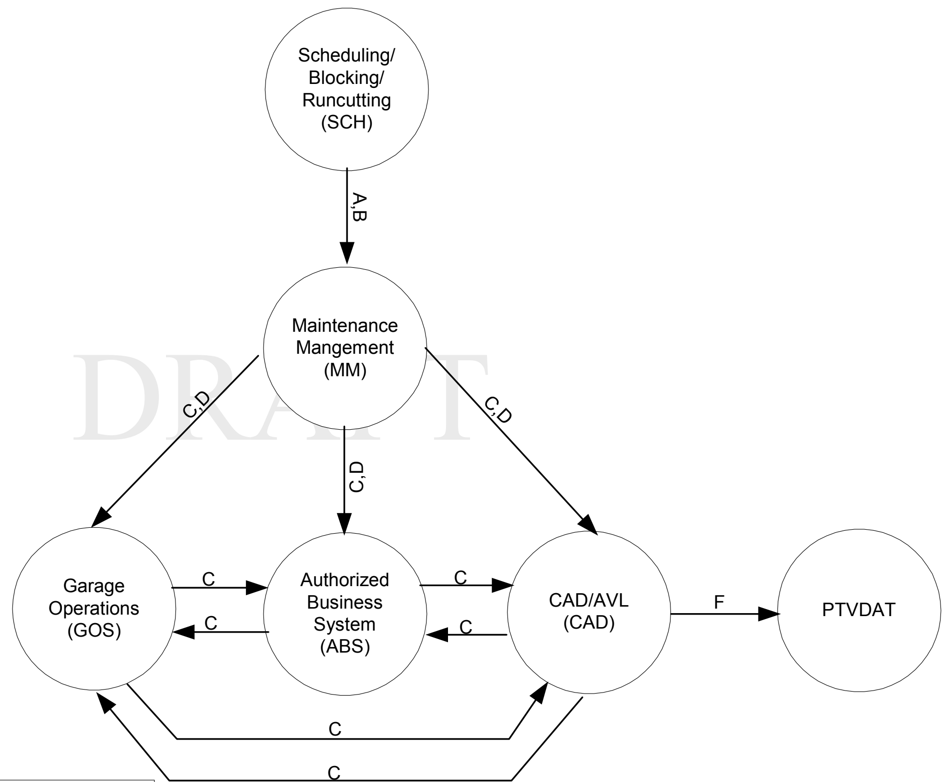
The Maintenance Management System (MM) maintains a history and schedule of PTV service. This provides the basis for maintenance planning, parts demand forecasts and vehicle availability planning, parts demand forecasts and vehicle availability planning. The Maintenance Management System also tracks PTV faults alarms and incidents or PTV service failures as input to the maintenance planning process.

The Maintenance Management System may also maintain the agency's PTV inventory, or this function may be performed by the Garage Operations System (GOS), Data Repository (DR), or other agency Authorized Business System (ABS).

5.7.1 Vehicle Assignments

The MM determines the list of PTVs that are expected to be available for service on a daily basis. Prior to the beginning of the service day the MM obtains the vehicle work assignments (blocks) for the business day. The MM then binds the available PTVs to the blocks or work to be performed, and provides these "bound" assignments to other business systems directly or through the Data Repository (DR) The MM also determines the service-available vehicles that have not been assigned for the day and provides those to other business systems.

The bound vehicle assignments may subsequently be changed as a result of unplanned events or incidents as the day progresses. The CAD/AVL system, Garage Operations System, or other Authorized Business System may change the vehicle assignment bindings by replacing a PTV that is not service ready with another PTV on the unassigned vehicles list. Figure 5.7.1 depicts the vehicle assignment binding dialog flows, with and without a Data Repository.

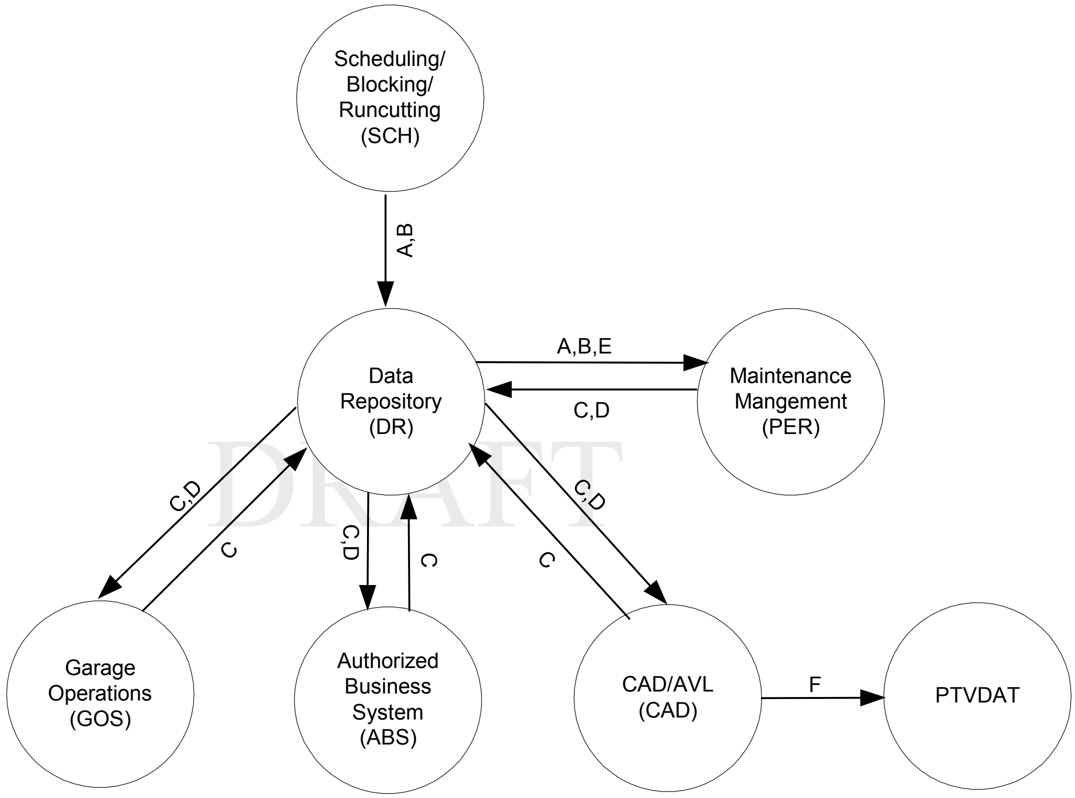


<u>Legend:</u>

A- Subscribe Timepoint List
Subscribe Pattern List
Subscribe Master Schedule Version
Subscribe Route Schedule
B- Subscribe Vehicle Assignments (Unbound)
C- Subscribe Vehicle Assignments (Bound)
D- Subscribe Unassigned Vehicles (Deprecated)
E- Subscribe Vehicle Itinerary
F- Command Change Assignments

Note: Schedule Distribution to GOS,ABS,CAD not depicted.

Figure 5.7.1.A
Vehicle Assignment Distribution Without Data Repository



Legend:

A- Subscribe Timepoint List
Subscribe Pattern List
Subscribe Master Schedule Version
Subscribe Route Schedule
B- Subscribe Vehicle Assignments (Unbound)
C- Subscribe Vehicle Assignments (Bound)
D- Subscribe Unassigned Vehicles (Deprecated)
E- Subscribe Vehicle Itinerary
F- Command Change Assignments

Note: Schedule Distribution to GOS,ABS,CAD not depicted.

Figure 5.7.1.B
Vehicle Assignment Distribution With Data Repository

5.7.2 Vehicle Health Monitoring

The health status of PTVs affects both current and planned service delivery. Accordingly this information is of vital interest to a variety of business systems. PTV health is reported in near-real time to the CAD/AVL System, and possibly other business systems based on agency architectures. PTV health information is also provided as part of the PTV's end of service day data unload. Figure 5.7.2 depicts the distribution of PTV health information. Note that it is possible for business systems to subscribe to health information directly from the PTV, however it is preferable to subscribe via the CAD/AVL System to conserve communications capacity.

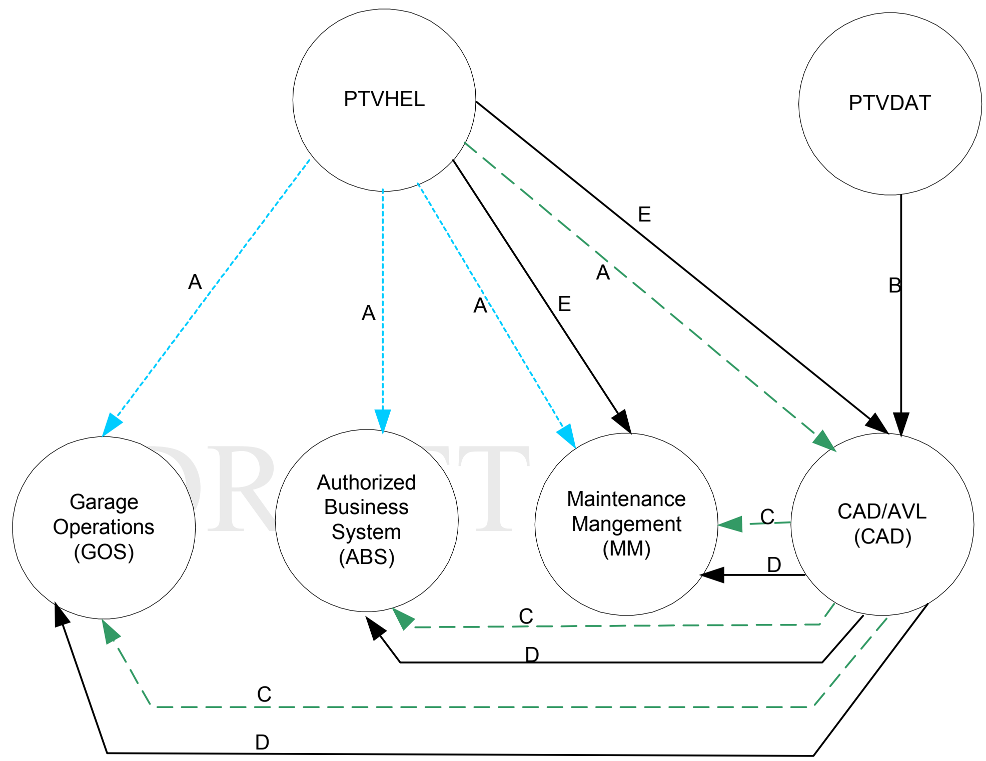


Figure 5.7.2.A
Vehicle Health Information Distribution
Without Data Repository

Legend:

A- Subscirbe PTV Health Alarms
B- Unload PTV Performance Data
C- Subscribe Fleet Health Alarms
D- Subscribe Daily Operating Data
E- Subscribe PTV Parameters

Preferred
Available

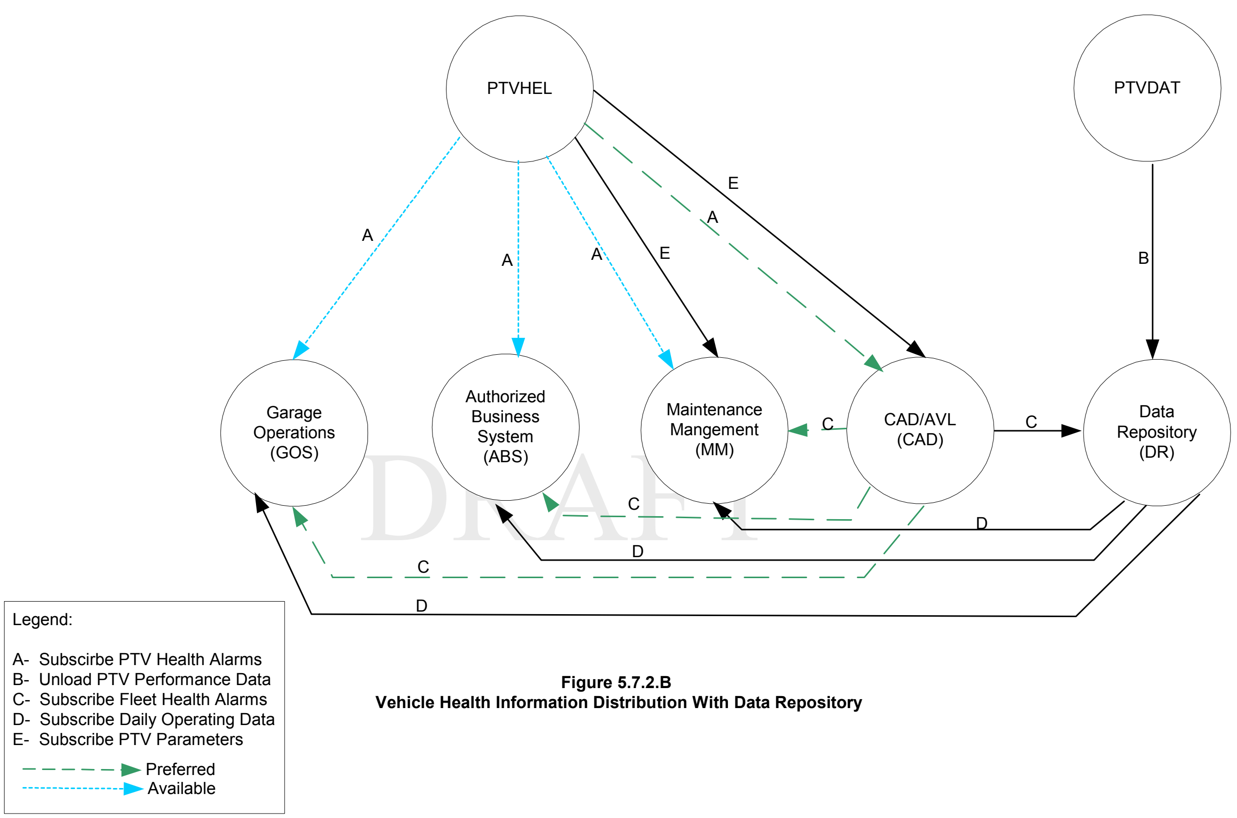


Figure 5.7.2.B
Vehicle Health Information Distribution With Data Repository

5.8 Customer Communications

The Customer Communications Business Process deals with all aspects of communicating with transit customers. This Includes assisting customers in determining the availability of transit services, providing transit literature (e.g. maps, guides for the handicapped) notifying the status of service, providing itineraries and providing customers with information while on the transit agency's property. Customer Communications are primarily supported by two business systems: Customer Service Systems (CSS) and Traveler Information Systems (TRV).

Customer Service systems support transit agency customer interactions via call centers and mail including:

- Itinerary creation for customer service call takers
- Mailing management for printed material
- Customer information profile management
- Complaint management and resolution
- Timetable creation
- Trip duration estimates based on scheduled running times
- Service bulletin creation and dissemination

Traveler Information Systems (TRV) support online interactions with customers via Passenger Information Displays, Traveler Internet Appliances, Kiosks, and Information Service Providers (ISP). Travelers can obtain the following via Internet Appliances, Kiosks, and ISPs:

- Nearest Transit service and parking information
- Transit Itineraries and Fares
- Landmark Locations
- Text timetables
- Stoppoint shelter and amenity information
- Transfer Opportunities
- Service Bulletins
- Maps

Agencies can provide a variety of information to customers via stoppoint and PTV-based Passenger Information Displays which may provide audible as well as visual announcements including:

- Next bus information at stoppoint
- Next stop information on PTVs
- Route and Destination information on PTVs
- Advertising and public service announcements on PTVs or stoppoints

Either the Traveler Information System (TRV) or Customer Service System (CSS) may serve as an Itinerary Generator for this other, or each may have its own internal itinerary generator. The CSS creates timetables from the schedule and provides those to the TRV. The TRV and/or the CSS may communicate with external Advanced Traveler Information Systems (ATIS) using SAE J2354 to obtain multimodal itineraries, itineraries including other transit agencies, weather information, etc, or to provide transit itineraries to external ATIS.

The TRV and CSS are provided fares for transit itineraries by the Fare System, and are provided on demand service status information by the CAD/AVL System. The CAD/AVL System also provides stoppoint ETA information which allows the TRV to provide next bus information at stoppoints. The CAD/AVL System provides service bulletins to CSS directly or via the Data Repository, similarly the Scheduling/Blocking/Runcutting System provides schedule information directly to CSS and TRV or via a Data Repository.

The CSS supports complaint resolution processes. Complaint investigations may require the investigator to review logs of vehicle movements from the CAD/AVL System, logs of operating data from PTVs or video images collected by cameras in PTVs or PTSFs and stored for later use.

Customer inquiries may use a variety of mechanisms to specify locations- particularly origins and destinations. Since customers generally do not know the latitude/longitude of their origin, destination or present location, the Subscribe Point Conversion dialog allows a business system to subscribe to translations of addresses, landmark names, etc. to latitude/longitude from another business system.

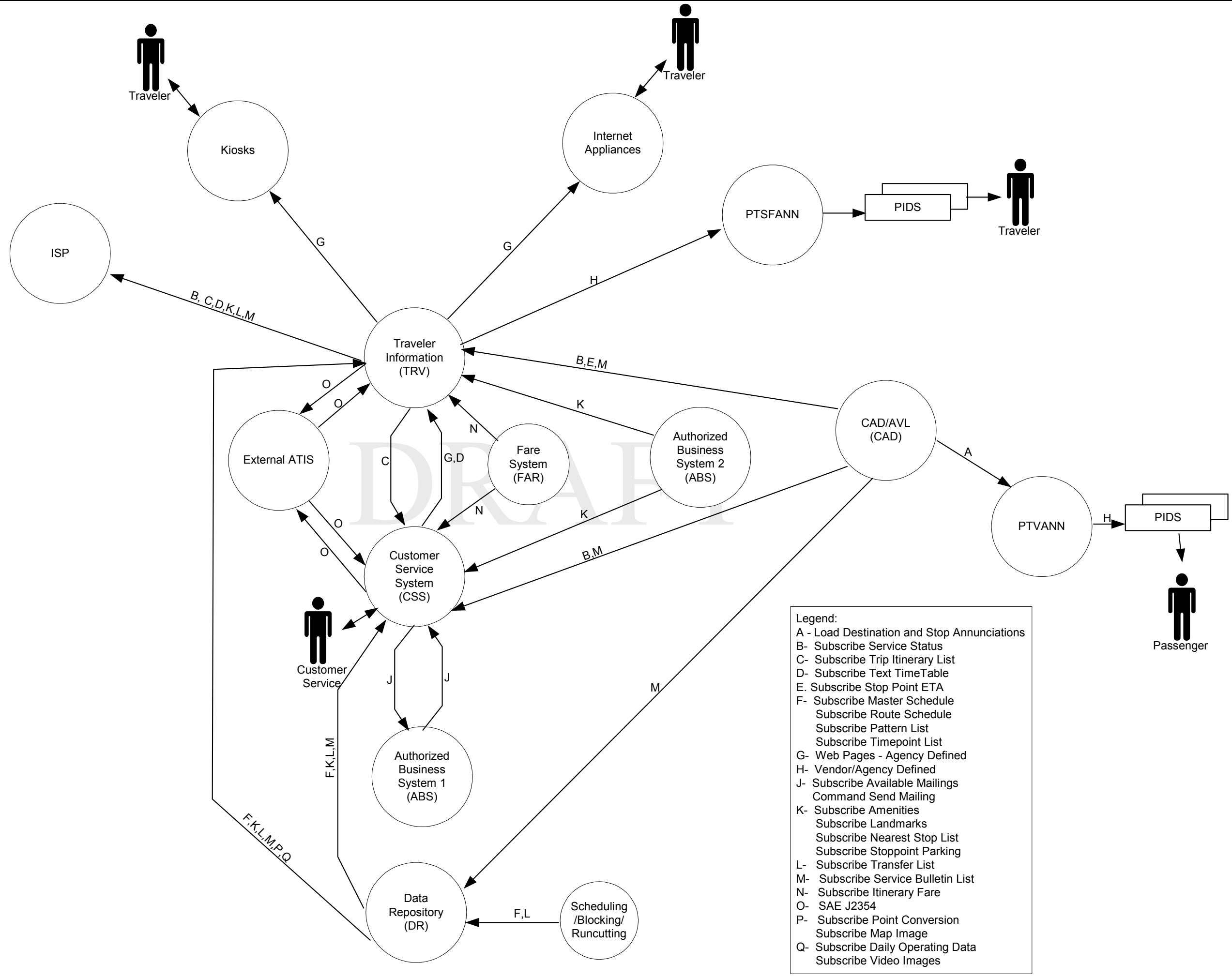


Figure 5.8.B
Customer Communications With Data Repository

5.9 Fare Collection

The fare collection business area defines information flows to support the operation of the onboard fare collection equipment. It does not encompass the definition of fare media formats, or interfaces to fare media. The fare collection business area provides at the following capabilities:

- Load Fare Related Operating Data
- Unload Fare Related History Data
- Report Onboard Fare Equipment Health Events
- Report Passenger Counts to Onboard Systems

5.9.1 Load Fare Related Operating Data

The Load Fare Collection Data dialog provides the load of operating data to the onboard fare collection equipment. Information that can be provided includes:

- Fare Zone Definitions
- Fare Definitions
- Stoppoint Information
- Day Type Definitions (holiday, weekday, etc.)

5.9.2 Unload Fare Related History Data

The Unload Fare Collection Data dialog provides the unload of operating data from the onboard fare collection equipment to the fixed fare collection application or Data Repository. Information that can be provided includes:

- Transaction History
- Smart Card Objects (defined by PAN/NJ RIS)
- Fare Collection Health Event Reporting
- Boarding/Alighting Counts by location

5.9.3 Report Onboard Fare Equipment Health Events

The "Subscribe Fare Collection Health" dialog provides a subscription –based fare collection equipment event and health reporting mechanism. This dialog provides information about fare equipment events in near-real time to allow for operational decision making at the transit control center, fare collection central facility or by other authorized parties. Note that similar information is available via the "Unload Fare Collection" dialog, however that information is not provided on a timely basis, and is intended for historical use.

5.9.4 Report Passenger Count to Onboard Systems

The "Subscribe Onboard Passenger Count" dialog provides the mechanism for an onboard system connected to the passenger counter sensors to report passenger count information to other onboard components and systems. Depending upon the configuration of the bus, this dialog can be used by the fare collection equipment to obtain passenger counts from another component by acting as a client. It can also be used by the fare collection equipment to provide passenger count information to another component by acting as the server.

5.9.5 Other Dialogs

The onboard fare collection equipment is likely to need information from other onboard components. TCIP provides the following dialogs from other business areas which may facilitate the

Subscribe Wireless LAN Status provides the status (available/unavailable) or the agency's wireless LAN. This status is useful in determining when to initiate file loads and unloads.

"Subscribe Operator Sign On" provides operator logon/logoff information.

"Subscribe Onboard Location" provides the vehicle's location and information on arrival at stoppoints and timepoints.

"Report Menu Selection" allows an external component to share the mobile data terminal and post messages and menus on the terminal, and obtain operator responses.

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,
- 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. The pattern does not define what is being subscribed to (or commanded, reported etc), but rather defines how to subscribe to (or command, report etc) data through a predefined dialog format. The subscription pattern provides a consistent framework for any query-type function. 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 File Transfer

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. TCIP message files contain one or more complete TCIP message instance(s), including attributes as well as elements. Text files are readable by virtually any type of computer system. TCIP dialogs are not required for file exchanges. Vendor-specific procedures govern the process for causing the files to be generated by one application and to be read by another application.

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 subscriber and a publisher. The publisher is the owner/producer/provider of information required by the subscriber who requires 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. If the publisher sends a CptSubErrorNotice at any time, the dialog immediately ends with no further data exchanges. Receipt of a CptSubErrorNotice, or a failure to communicate, or to receive an expected response must be recovered by a vendor specified process within the component (subscriber or publisher).

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 subscriber requests the information and the publisher provides it on a one-time basis.
- The second form is the periodic subscription. The subscriber requests the information and the publisher provides it initially and on a recurring basis at intervals specified in the subscription request.
- The third form is the event-driven subscription. The subscriber requests the information and the publisher 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 subscriber to cancel a periodic or event-driven subscription. In the event that the publisher receives a subscription request message (AaaXxxSub) with a subscription header data frame indicating that the subscription is to be cancelled, the publisher 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 and the publisher 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. Agencies will need to coordinate with other agencies with whom they share data to ensure uniqueness of these identifiers (e.g. within a metropolitan area).

The CPT-SubscriptionHeader provides an identifier for each subscription of type CPT-RequestIdentifier. This identifier allows multiple subscriptions between a subscriber-publisher pair to exist simultaneously and operate independently.

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

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

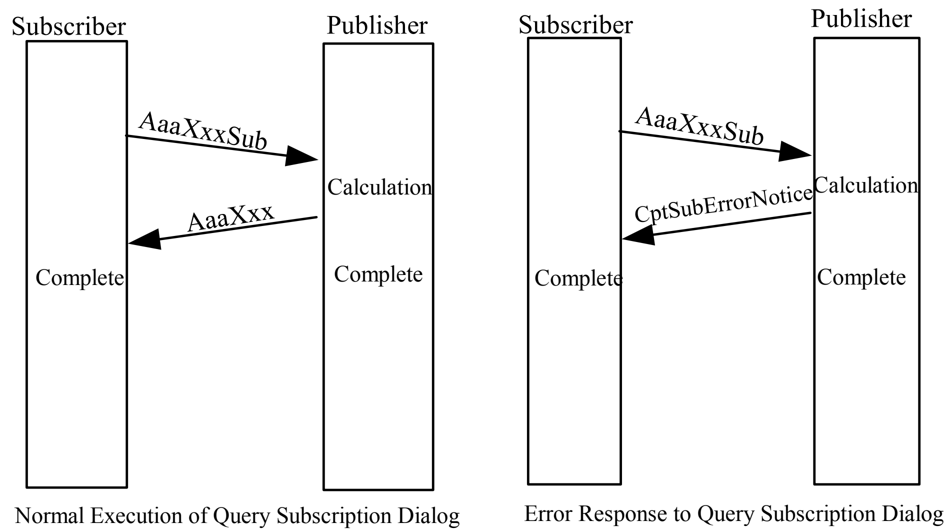


Figure 7.1.A
Query Version of the Subscription Dialog Pattern

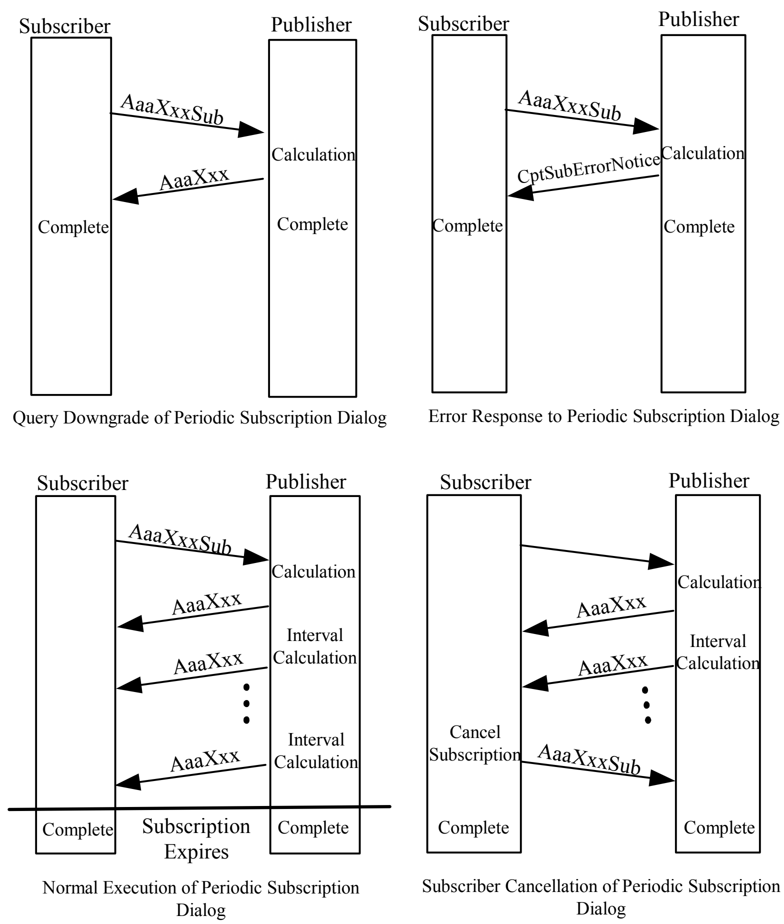


Figure 7.1.B
Periodic Version of the Subscription Dialog Pattern

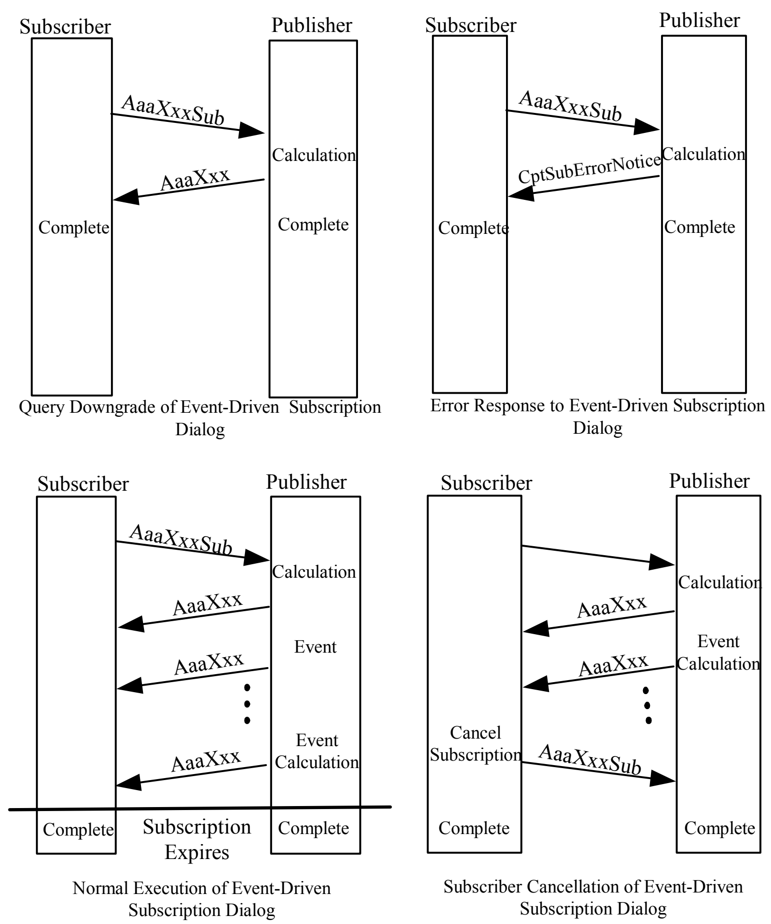


Figure 7.1.C
Event Driven Version of the Subscription Dialog Pattern

Although the subscriber requests the form of the subscription in CPT-SubscriptionHeader, the publisher may change (downgrade) the request in the response. The allowed changes by the publisher 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.

The publisher notifies the subscriber of the downgrade by updating the fields from the subscription requests that are replicated in the response message.

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 publisher 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 from a CAD/AVL System, the agency may require the CAD/AVL system provider to service the periodic subscription, and not downgrade it to a query.

Some subscription dialogs are used to exchange large data sets usually expressed as lists that may have small, but frequent changes. Such dialogs may implement query or event-based "row versioning". In query-based row versioning, the subscriber sends a query with a field indicating that only data that has changes since a specified date & time should be included in the response from the publisher. In event-based row versioning, the initial query elicits the entire dataset, however updates to the dataset trigger event messages to the subscriber indicating that the message contains only updates since a specified date & time. In either case the message to the subscriber may contain a list of deleted items as well as a list of new or updated items in the list.

In some cases a single subscription dialog is reused a number of times within an agency architecture. An example is when more than one application makes updates to the same information. In such cases, event-based subscriptions typically are used to exchange updates among business systems. Figure 7.1.D depicts and exchange network. In this example dialog X is used to transfer a version of some information to business systems B, C and D from the originator – business system A. Event-driven subscriptions allow B,C and D to exchange row updates to the data. Note that B,C and D must only generate and event and send an update when they generate the change. "Relaying" changes will result in a potential for an undesired event explosion.

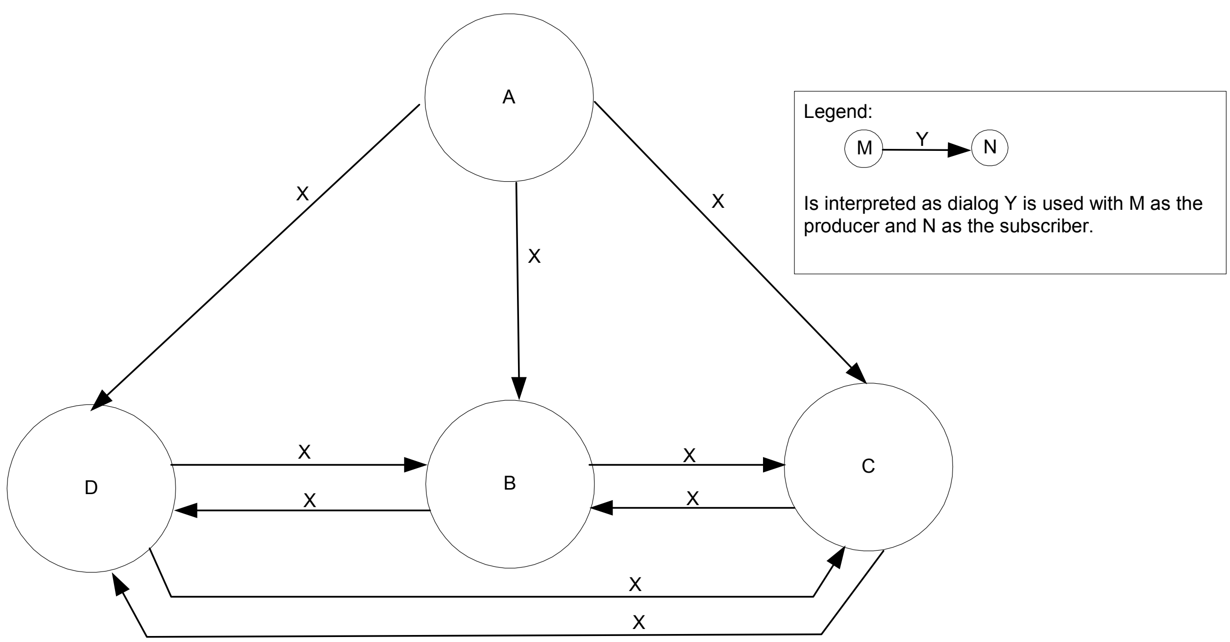


Figure 7.1.D
Using Event-Driven Subscription Dialogs to Distribute Updates to Agency Data

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 execution of the Command Response dialog pattern.

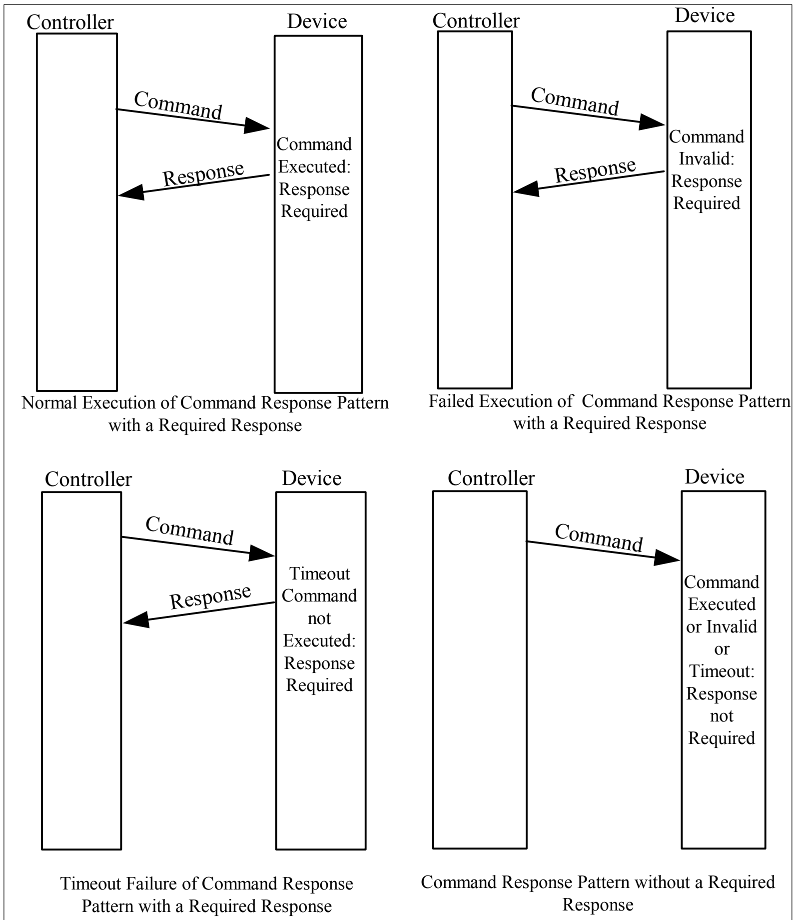


Figure 7.2
Command Response Dialog Pattern

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 Report dialog pattern.

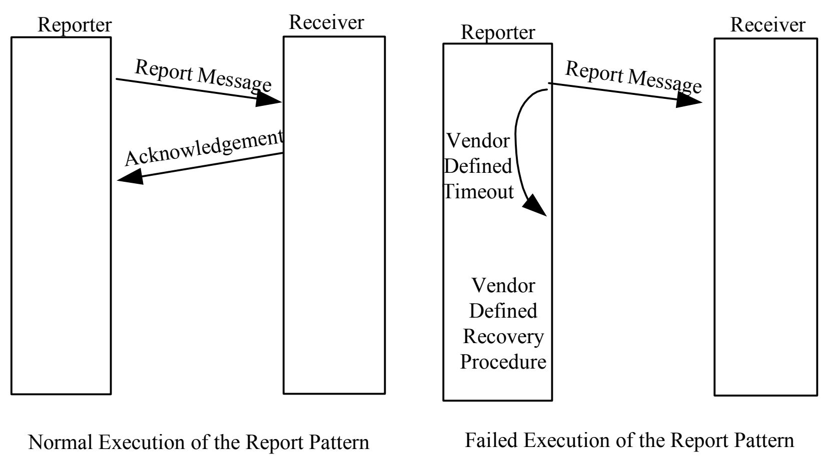


Figure 7.3
Report Dialog Pattern

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.

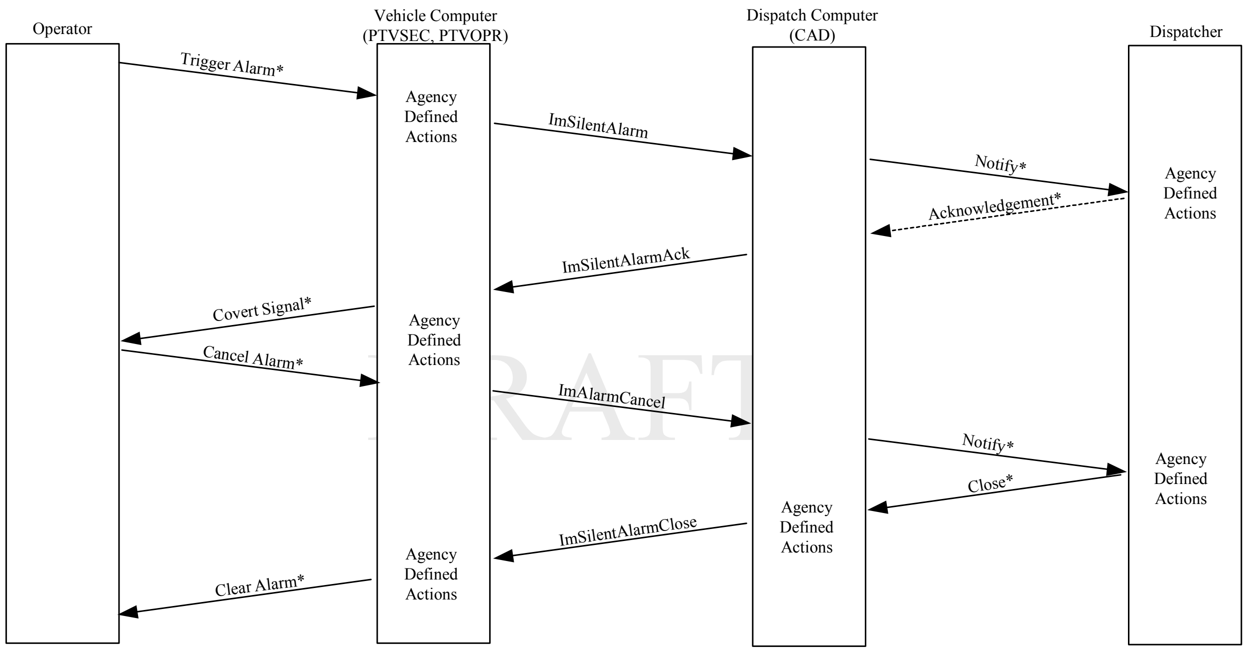
After an agency/vendor defined notification and acknowledgment between the dispatch computer and the dispatcher, the dispatch computer provides an acknowledgment (ImSilentAlarmAck) to the vehicle. The dispatcher and/or the dispatch computer may initiate other vendor/agency defined actions as a result 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.



*Agency/Vendor defined transactions

Figure 7.4
Normal Execution of the Silent Alarm Dialog

7.5 Load Pattern

This pattern is intended to load large files from a fixed business system to a corresponding onboard component. Files may include configuration information, software applications or other information. Loads may be performed using a laptop computer, or a wireless LAN or other network connection. Loads may also be performed using portable media such as flash cards or CDs (batch mode). This dialog pattern is applicable to loads performed over a network connection or a laptop plugged into the onboard/field component. Normally loads 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 loaded. The fixed component is responsible for keeping track of the correct version number(s) of the loaded file(s) as appropriate for the onboard/field 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. Similarly, the correct version number for a field component may be the same for all stoppoints, field locations etc., or may be tailored on a site by site or component by component basis.

The onboard or field component is responsible for keeping track of the version number(s) of the load files it has stored. Onboard or field components may or may not keep multiple versions of the same load file depending on internal memory size and file management capabilities. Manufacturers of onboard or field 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 or field component remains usable.

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

The onboard or field component will initiate a load upon:

- Receipt of a CptForceLoad 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 load request interval (see below) has elapsed since the last load initiation.
- Determination that the maximum load request interval (see below) has expired and the wireless LAN is available.

The minimum load request interval is a parameter that prevents the onboard components from continuously initiating loads 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 load request. This interval is overridden by receipt of a CptForceLoad message. The value of this parameter is locally defined. A recommended initial default interval is 60 minutes.

The maximum load request interval is a parameter that prevents the onboard or field components from failing to initiate new loads 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 or field components failing to obtain new loads. If the component remains in wireless LAN coverage for a period exceeding this interval, the component will initiate a new load. The value of the maximum load request interval is locally defined. A recommended initial default interval is 720 minutes (12 hours).

The dialog pattern executes as follows:

1. The onboard or field 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 load files stored in the onboard or field component.

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

3. The onboard or field 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 or field component should have (e.g. farebox doesn't use VLU software files).

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

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

6. Upon receipt of the file the onboard or field 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 or field component sends a CptLoadControl 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 Load dialog pattern.

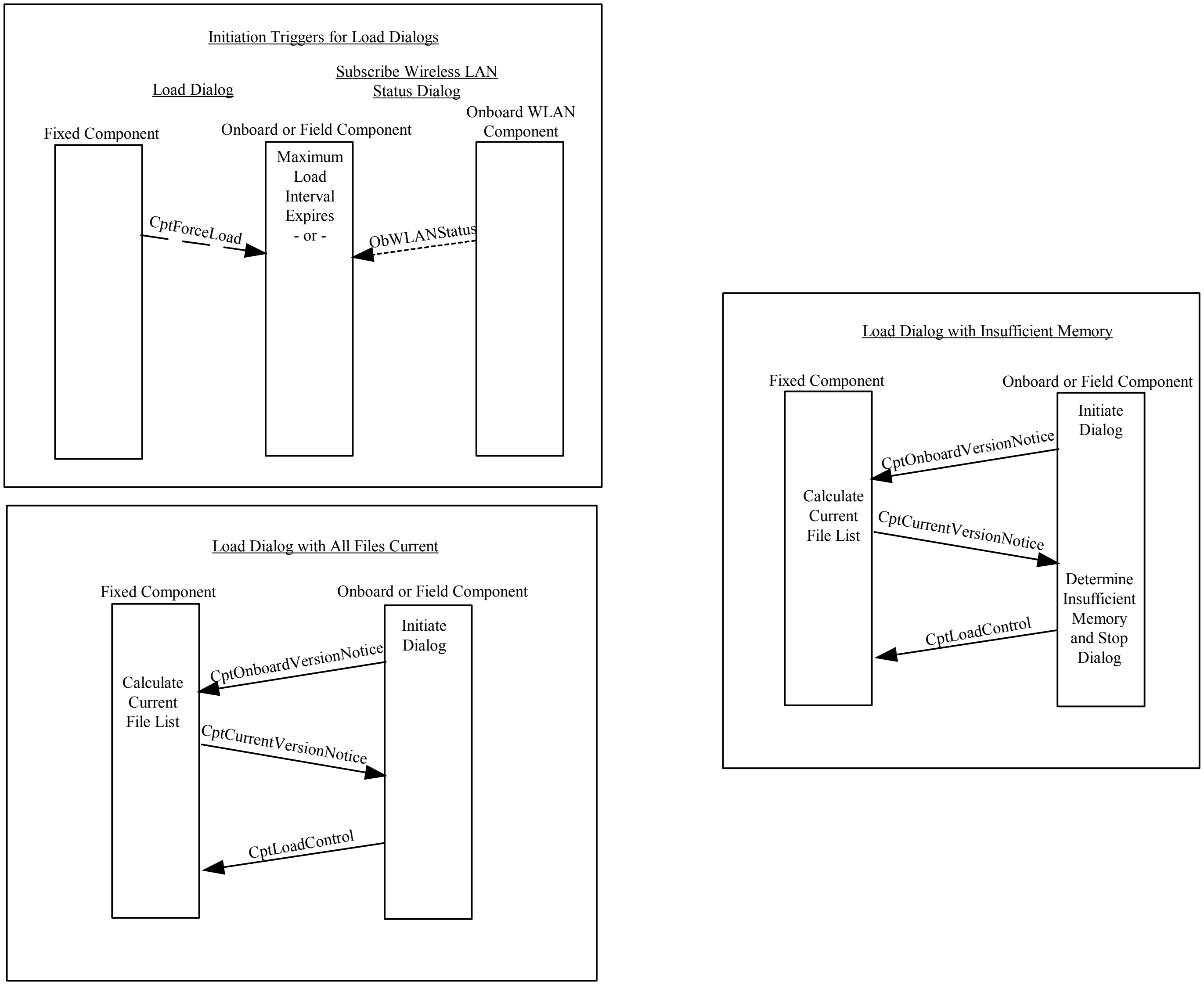


Figure 7.5
Depicts the Execution of the Load Dialog Pattern

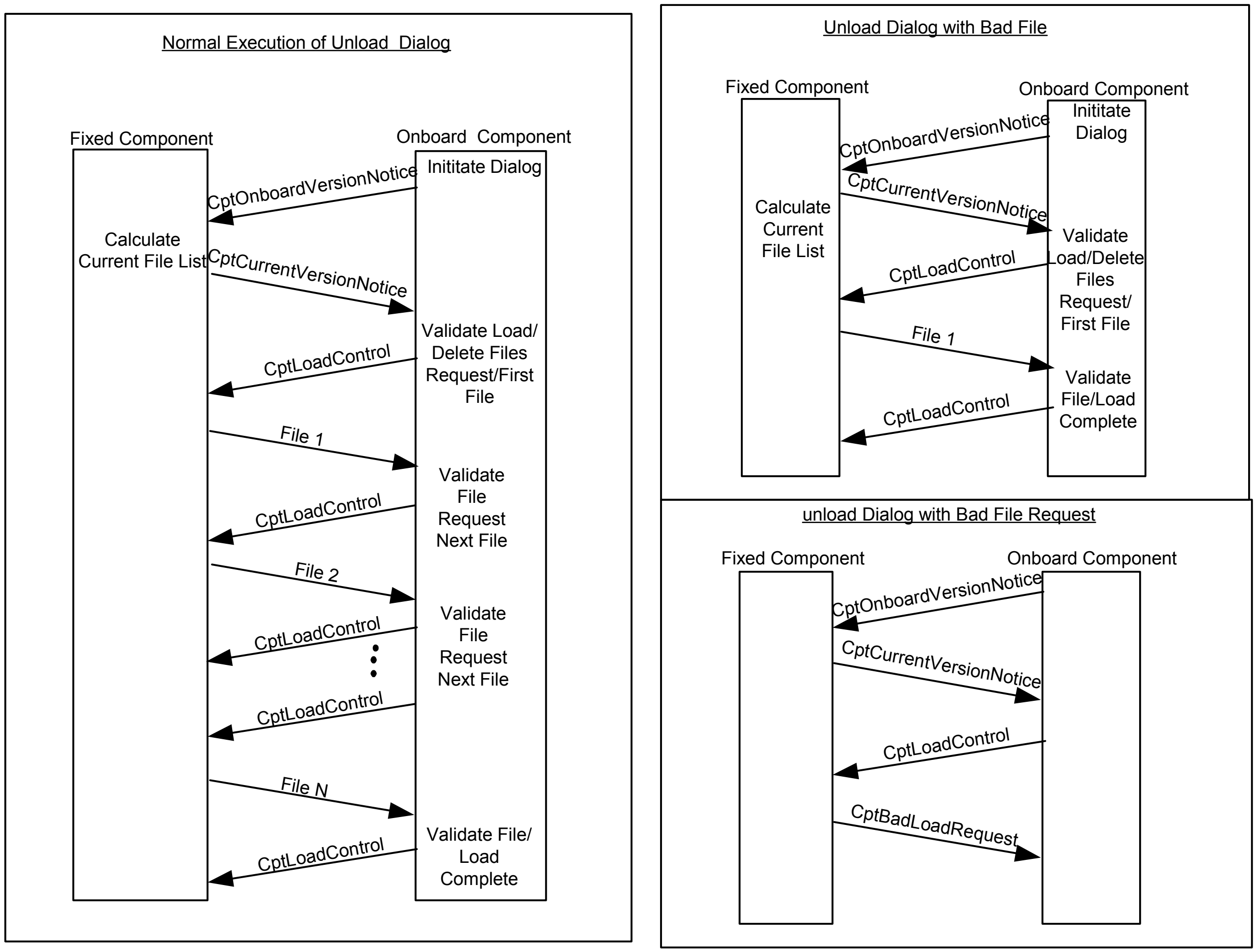


Figure 7.6.A
Depicts the Execution of the Unload Dialog Pattern

7.6 Unload Pattern

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

The fixed component (laptop or fixed business system) is responsible for maintaining configuration control over the unloaded files. The onboard or field component is responsible for storing the files to be unloaded until the file is listed for deletion in a CptUnloadControl message or until the onboard or field 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 unload initiation is always performed by the mobile component by sending a CptFilesToUnload message to the fixed component. Note that a unload initiation may not result in an actual file transfer.

The onboard or field component will initiate a unload 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 unloaded.
- Determination the wireless LAN is available, and a new file has become ready to unload.
- Receipt of a CptForceUnload message from the fixed component. This message exists primarily to trigger the unload to a laptop plugged into the onboard or field system(s).

The dialog pattern executes as follows:

1. The onboard or field component initiates the dialog based on one of the 3 criteria described above and sends a CptFilesToUnload 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 or field component awaiting a unload.
2. The fixed component upon receipt of the CptFilesToUnload message sends a CptUnloadControl message to the onboard or field component. This message informs the onboard or field component of the next file to unload (if applicable) and any files to be deleted.
3. The onboard or field component determines:

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

   B. Whether any files are to be deleted without unloading them. For example the fixed component may have successfully received an unload on a previous attempt, but been unable to notify the onboard or field component due to a loss of WLAN coverage.
4. The onboard or field component deletes any files specified to be deleted, and if an unload 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 fixed component, and is not on hand in the onboard or field component, the deletion request is ignored.
6. If a file is specified for unloading by the fixed component and the file is not on hand, the onboard or field component sends a CptUnloadRequestError message to the fixed component and terminates the dialog. The onboard or field component may re-initiate the dialog immediately if the WLAN is still available and there are still files to unload.
7. Upon receipt of an unloaded file, the fixed component sends a CptUnloadControl message to the onboard or field component, and the dialog goes to step 3 above. The fixed end is responsible for including the successfully unloaded file in the files to delete section of the CptUnloadControl message.

Figure 7.6 depicts the execution of the unload dialog pattern.

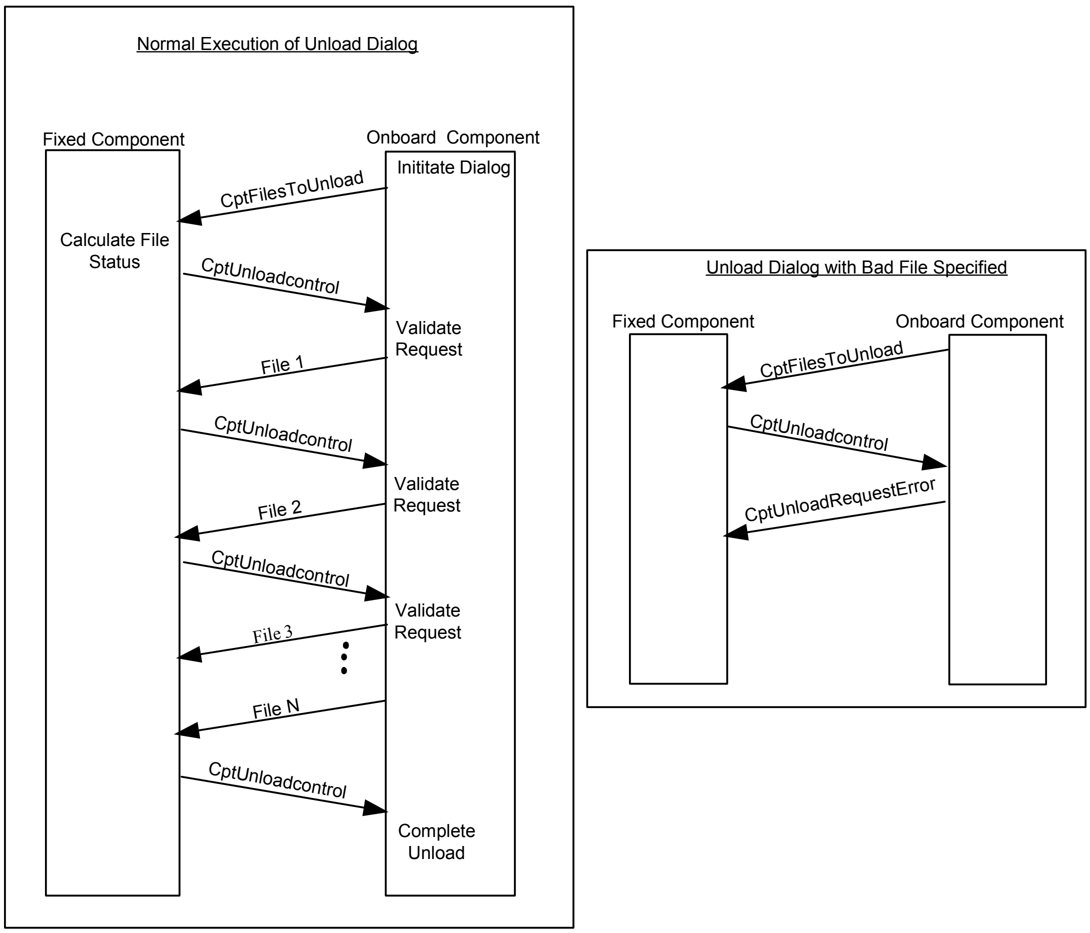


Figure 7.6.B
Unload Dialog Pattern

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.

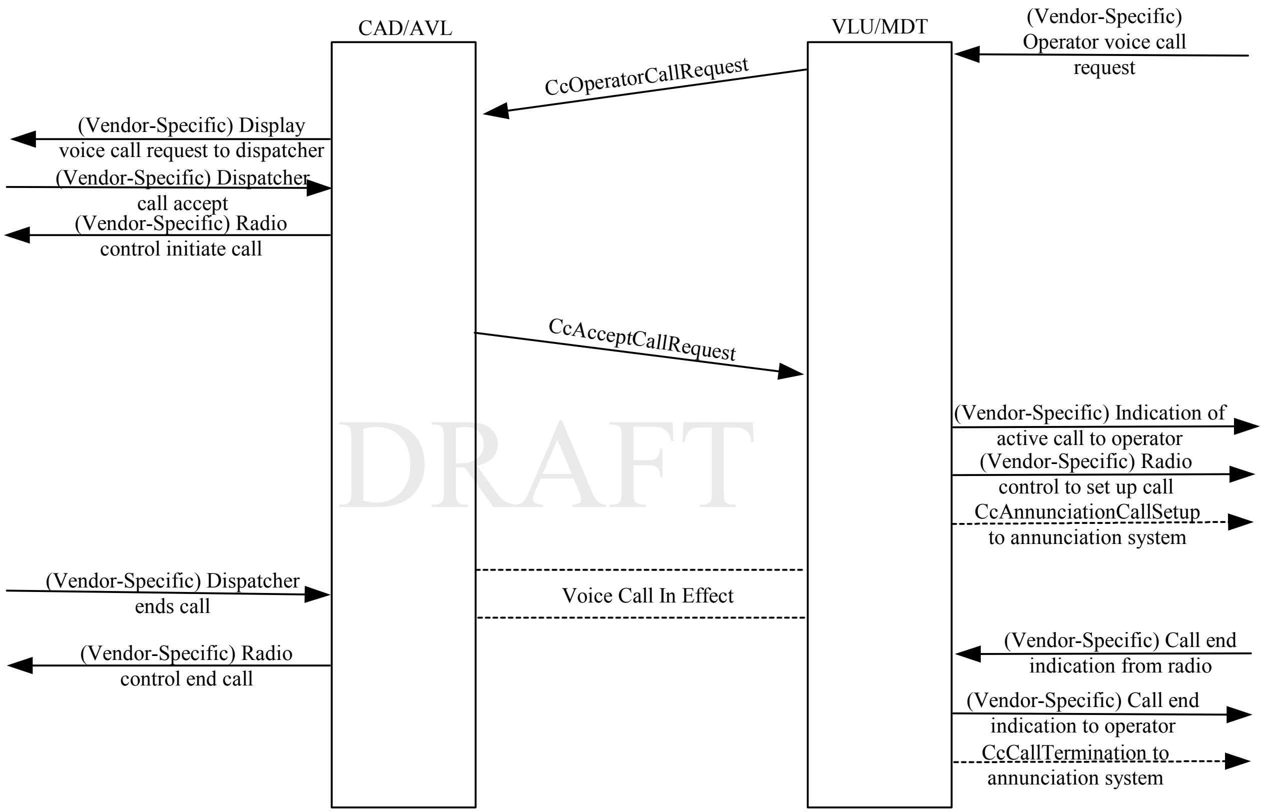


Figure 7.7.1
A Operator-Initiated Voice Radio Call Pattern (Accepted)

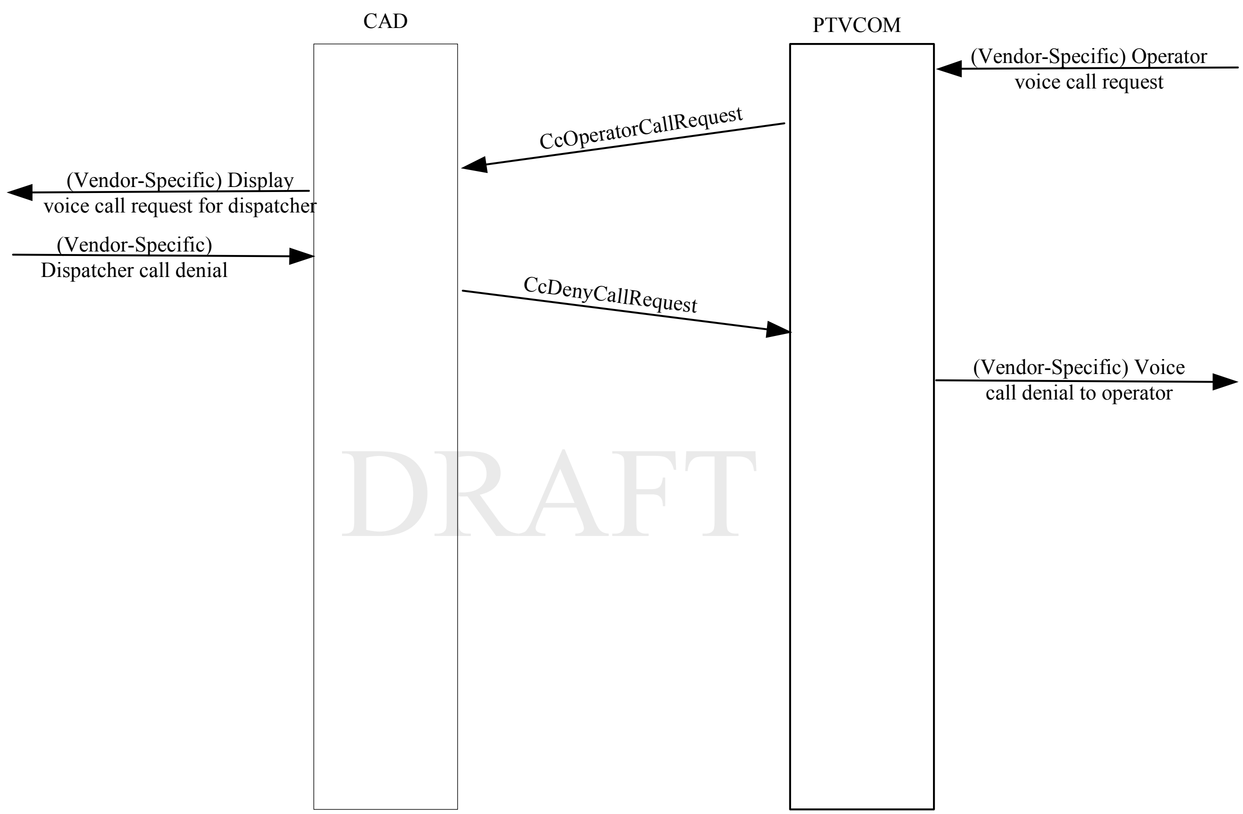


Figure 7.7.1 B
Operator Initiated Voice Radio Call (Denied)

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.

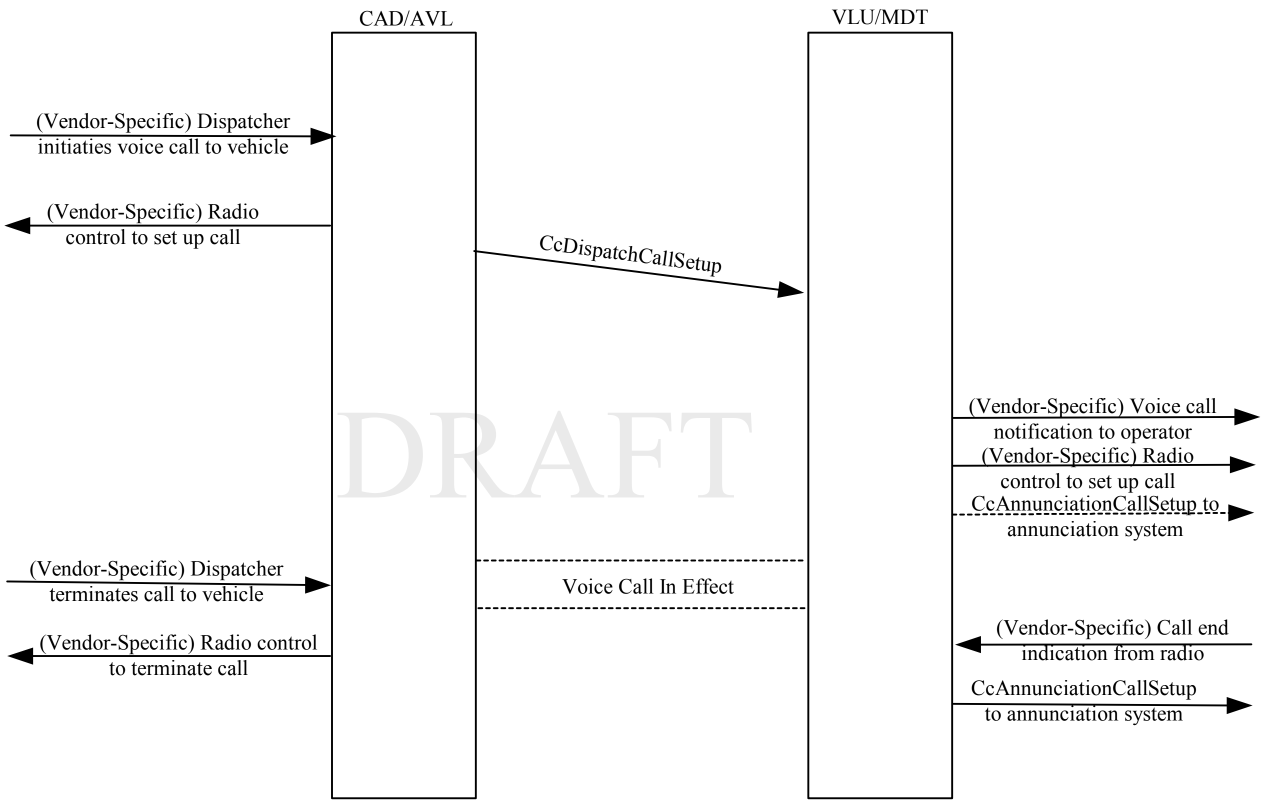


Figure 7.7.2
Dispatcher-Initiated Voice Radio Call Pattern

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.

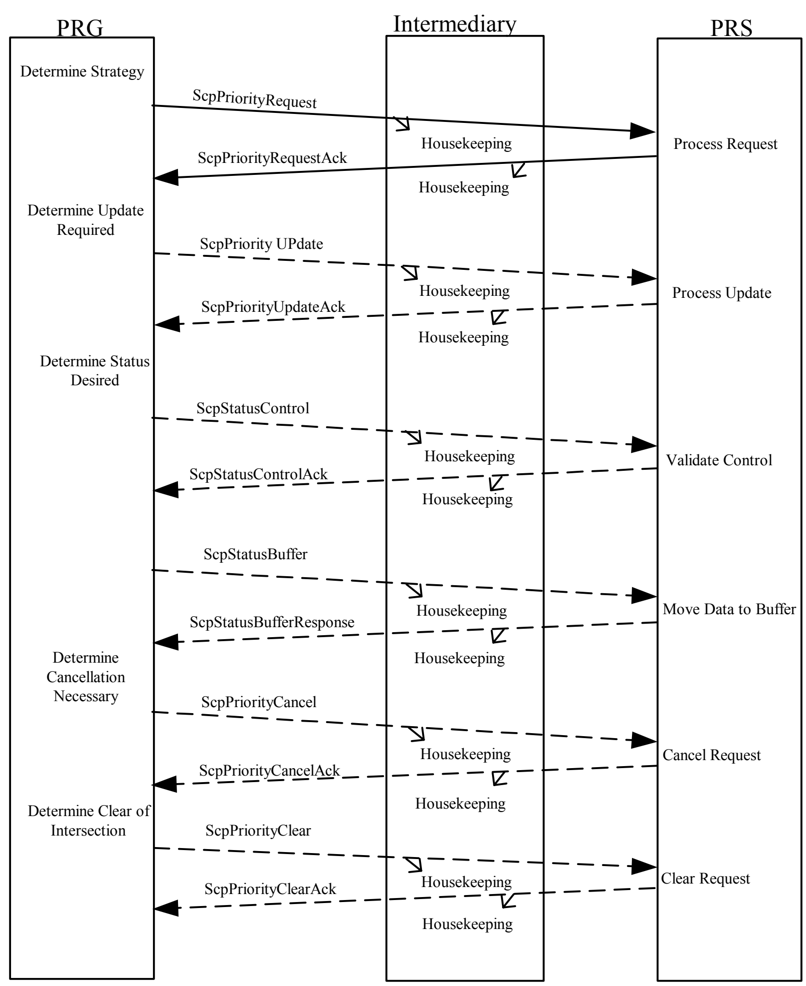


Figure 7.8
Signal Control & Prioritization Dialog Pattern

7.9 Blind Notification Pattern

This is a one message pattern that provides for a notifier to send information to a receiver, without an expected response. The notifier does not implement any error recovery if the message is not delivered successfully.

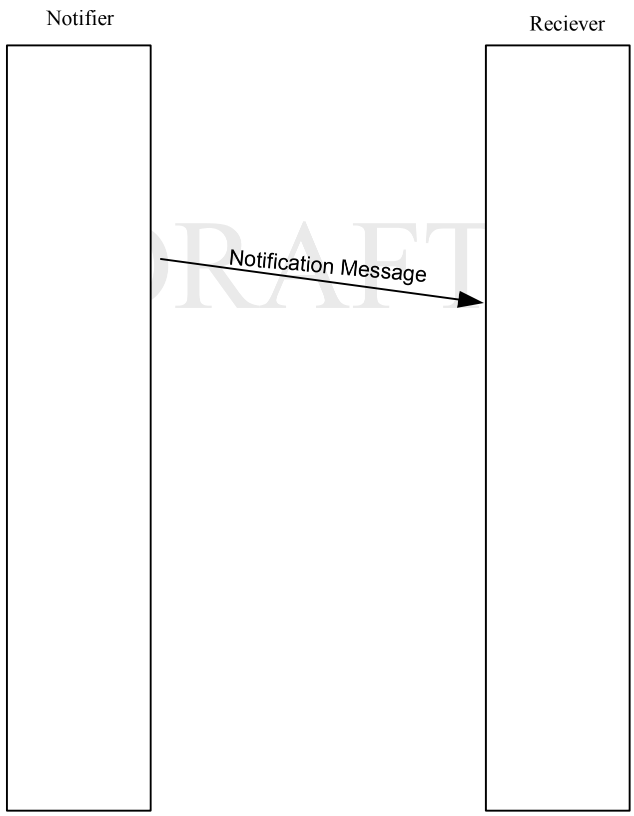


Figure 7.9
Blind Notification Pattern

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 other 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.

To make TCIP data easier to understand, naming conventions have been adopted for TCIP Data Elements, Data Frames, and Messages. Data Elements are identified by a Business Area Name in all capital letters, followed by a hyphen, followed by the name of the data. For example, CPT-EmployeeFirstName. Data Frames are identified by a Business Area Name in all capital letters, with no hyphen, followed by the name of the data. For example, SCHPatternInfo. Messages are identified by a Business Area Name with an initial capital letter, followed by the name of the data. For example, PiNearestStopListSub.

In some cases TCIP imports data frame of data element definitions from other standards such as SAE J2266 Location Referencing Message Specification (LRMS). In these cases, the ASN.1 usage is to identify the data element or data frame using a dot notation with the source specification to the left of the period in capital letters and the name as defined in the source specification to the right. For example, LRMS.GeoLocation. In the SML schema, the equivalent usage is a colon notation with the source name in lower case, for example, lrms:GeoLocation.

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 normative 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 E. To be conformant implementation must comply with the XML requirements in Annexes A, B and C, as well as the XML Schema in Annexes E.

Commentary: SAE J2630 "Converting ATIS message Standards from ASN.1 to XML" was used as a guideline when converting from ASN.1 to XML.

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.

TCIP also references other Intelligent Transportation System Standards in defining data elements, data frames and messages. These standards are:

- Location Referencing Message Standard (LRMS) defined by SAE J2266
- Advanced Traveler Information System Standard (ATIS) defined by SAE J2354
- Traffic Management Data Dictionary (TMDD) defined by Joint ITE/AASHTO Standard Number TM1.03
- International Traveler Information Systems (ITIS) Phrase Lists defined by SAE J2540
- Object Definitions for Signal Control & Priority defined by NTCIP 1211

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 8.1.2.1.1 and in 8.1.2.1.3.
UTF8String	This base type is used to create several TCIP string subtypes in section 8.1.2.1.3.
Numeric String	This base type is similar to a UTF8String, but limited to numeric values. It is used to create base types in section 8.1.2.1.3.
OCTET String	This base type is used to create base types in section 8.1.2.1.3.

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 ULONG. These subtypes are defined as follows:

Table 8.1.2.1.1		
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 (-2,147,483,647 .. 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…235960999. 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. Seconds are allowed to equal 60 to allow for a leap second.
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.

DATETIME	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 String, Name and Identifier Subtypes

These subtypes are identified as follows:

Table 8.1.2.1.3

ASN.1 Definition	Description	XML Types	Narrowband Encoding
TELEPHONE ::= NUMERIC STRING (SIZE(1..16))	A string of up to 16 digits to dial	xs:string	One octet length field followed by 1-16 characters defining digits to dial.
FOOTNOTE ::=UTF8String (SIZE(1…255))	Footnote is a memo field for defining comments and other free form text.	xs:string	One or two octet length field followed by 1-255 characters.
MEMSHORT16 ::= OCTET STRING (SIZE(1..16))	A binary field of up to 16 octets.	xs:base64binary	One octet length field followed by 1-16 octets of data.
MEMSHORT2 ::= OCTET STRING (SIZE(1..2))	A binary field of up to 2 octets.	xs:base64binary	One octet length field followed by 1-2 octets of data.
MEMSHORT23 ::= OCTET STRING (SIZE 1..23))	A binary field of up to 23 octets.	xs:base64binary	One octet length field followed by 1-23 octets of data.
MEMSHORT32 ::= OCTET STRING (SIZE(1..32))	A binary field of up to 32 octets.	xs:base64binary	One octet length field followed by 1-32 octets of data.
MEMVSHORT ::= OCTET STRING (SIZE(1..255))	A binary field of up to 255 octets.	xs:base64binary	One or two octet length field followed by 1-255 octets of data.

Table 8.1.2.1.3			
ASN.1 Definition	**Description**	**XML Types**	**Narrowband Encoding**
MEMSHORT ::=OCTET STRING (SIZE(1..2047))	A binary field of up to 2047 octets.	xs:base64binary	One or two length octet field followed by 1-2047 octets of data.
MEMLONG ::= OCTET STRING (SIZE(1..2,000,000)	A binary field of up to 2,000,000 octets.	xs:base64binary	One to three octet length field followed by 1-2,000,000 octets of data.
NAME1 ::=UTF8String (SIZE (1))	A string of 1 character.	xs:string	One octet length field followed by 1 character of data.
NAME2 ::=UTF8String (SIZE(1..2))	A string of 2 characters.	xs:string	One octet length field followed by up to 2 characters of data.
NAME3 ::=UTF8String (SIZE (1..3))	A string of 3 characters.	xs:string	One octet length field followed by up to 2 characters of data.
NAME4 ::=UTF8String (SIZE (1..4))	A string of 4 characters.	xs:string	One octet length field followed by up to 2 characters of data.
NAME5 ::=UTF8String (SIZE (1..5))	A string of 5 characters.	xs:string	One octet length field followed by up to 2 characters of data.
NAME6 ::=UTF8String (SIZE (1..6))	A string of 6 characters.	xs:string	One octet length field followed by up to 2 characters of data.
NAME7 ::=UTF8String (SIZE (1..7))	A string of 7 characters.	xs:string	One octet length field followed by up to 2 characters of data.
NAME8 ::= UTF8String (SIZE(1..8))	A string of 8 characters.	xs:string	One octet length field followed by up to 2 characters of data.
NAME9 ::=UTF8String (SIZE (1..9))	A string of 9 characters.	xs:string	One octet length field followed by up to 2 characters of data.
NAME10 ::=UTF8String (SIZE (1..10))	A string of 10 characters.	xs:string	One octet length field followed by up to 2 characters of data.
NAME11 ::=UTF8String (SIZE (1..11))	A string of 11 characters.	xs:string	One octet length field followed by up to 2 characters of data.

Table 8.1.2.1.3			
ASN.1 Definition	**Description**	**XML Types**	**Narrowband Encoding**
NAME12 ::=UTF8String (SIZE (1..12))	A string of 12 characters.	xs:string	One octet length field followed by up to 2 characters of data.
NAME13 ::=UTF8String (SIZE (1..13))	A string of 13 characters.	xs:string	One octet length field followed by up to 2 characters of data.
NAME14 ::=UTF8String (SIZE (1..14))	A string of 14 characters.	xs:string	One octet length field followed by up to 2 characters of data.
NAME15 ::=UTF8String (SIZE (1..15))	A string of 15 characters.	xs:string	One octet length field followed by up to 15 characters of data.
NAME16 ::= UTF8 String (SIZE(1…16))	A string of 16 characters.	xs:string	One octet length field followed by up to 16 characters of data.
NAME17 ::=UTF8String (SIZE (1..17))	A string of 17 characters.	xs:string	One octet length field followed by up to 17 characters of data.
NAME18 ::=UTF8String (SIZE (1..18))	A string of 18 characters.	xs:string	One octet length field followed by up to 18 characters of data.
NAME19 ::=UTF8String (SIZE (1..19))	A string of 19 characters.	xs:string	One octet length field followed by up to 19 characters of data.
NAME20 ::=UTF8String (SIZE (1..20))	A string of 20 characters.	xs:string	One octet length field followed by up to 20 characters of data.
NAME21 ::=UTF8String (SIZE (1..21))	A string of 21 characters.	xs:string	One octet length field followed by up to 21 characters of data.
NAME22 ::=UTF8String (SIZE (1..22))	A string of 22 characters.	xs:string	One octet length field followed by up to 22 characters of data.
NAME23 ::=UTF8String (SIZE (1..23))	A string of 23 characters.	xs:string	One octet length field followed by up to 23 characters of data.

Table 8.1.2.1.3			
ASN.1 Definition	**Description**	**XML Types**	**Narrowband Encoding**
NAME24 ::=UTF8String (SIZE (1..24))	A string of 24 characters.	xs:string	One octet length field followed by up to 24 characters of data.
NAME25 ::=UTF8String (SIZE (1..25))	A string of 25 characters.	xs:string	One octet length field followed by up to 25 characters of data.
NAME30 ::= UTF8String (SIZE(1..30))	A string of 30 characters.	xs:string	One octet length field followed by up to 30 characters of data.
NAME32 ::=UTF8String (SIZE(1..32))	A string of 32 characters.	xs:string	One octet length field followed by up to 32 characters of data.
NAME40 ::=UTF8String (SIZE (1..40))	A string of 40 characters.	xs:string	One octet length field followed by up to 40 characters of data.
NAME60 ::=UTF8String (SIZE (1..60))	A string of 60 characters.	xs:string	One octet length field followed by up to 60 characters of data.
IDENS ::= USHORT	An identifier 16 bits in length	xs:unsignedshort	A two octet binary unsigned numeric field.
IDENL ::= ULONG	An identifier 32 bits in length	xs:unsignedlong	A four octet binary unsigned numeric field.
TEXTLONG ::=UTF8String (SIZE(1..50000))	A field for conveying large text sequences.	xs:string	1-3 octet length filed followed by 1050,000 character text field.

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.

TCIP data elements are based on the ASN.1 types defined in 8.1.1 above and the TCIP subtypes defined in 8.1.2 above. Example Section 8.1.2 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 other data frames) into 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. Many 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. 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

```
SCHPatternInfo::=SEQUENCE {
patternID              SCH-PatternID,
pattern-designator     SCH-PatternDesignator                      OPTIONAL,
pattern-name           SCH-PatternName                            OPTIONAL,
pattern-notes          SEQUENCE (SIZE(1..100)) OF SCH-NoteID      OPTIONAL,
triggers               SEQUENCE (SIZE(1..200)) OF SCH-ActivationID OPTIONAL,
mode                   CPT-Mode                                   OPTIONAL,
segments               SEQUENCE (SIZE(1..100)) OF SCH-PatternSegmentID,
radio-zones            SEQUENCE (SIZE(1..100)) OF CPTRadioZone    OPTIONAL
}
```

Included Data Frame

Included Data Elements

Example Data Frame Definition in XML

```
<xs:complexType name="SCHPatternInfo" >
    <xs:sequence>
      <xs:element name="pattern-designator" type="SCH-PatternDesignator"  minOccurs="0"/>
      <xs:element name="pattern-name" type="SCH-PatternName"  minOccurs="0"/>
      <xs:element name="pattern-notes"  minOccurs="0">
        <xs:complexType>
          <xs:sequence minOccurs="1" maxOccurs="100">
            <xs:element name="pattern-note" type="SCH-NoteID" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="triggers"  minOccurs="0">
        <xs:complexType>
          <xs:sequence minOccurs="1" maxOccurs="200">
            <xs:element name="trigger" type="SCH-ActivationID" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="mode" type="CPT-Mode"  minOccurs="0"/>
      <xs:element name="segments" >
        <xs:complexType>
          <xs:sequence minOccurs="1" maxOccurs="100">
            <xs:element name="segment" type="SCH-PatternSegmentID" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
      <xs:element name="radio-zones"  minOccurs="0">
        <xs:complexType>
          <xs:sequence minOccurs="1" maxOccurs="100">
            <xs:element name="radio-zone" type="CPTRadioZone" />
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
</xs:complexType>
```

Figure 8.2

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 Element

```
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 to transfer files on removable media such as smart cards, compact disks, or diskettes (conformance class 2A). 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 (conformance class 1A). 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 via the PTVCOM entity for notifying the mobile equipment when the link is, and is not, available using the "Subscribe Wireless LAN Status" dialog. This allows subscribing components or applications onboard the vehicle to initiate loads or unloads 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 extensible 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. Note that while every message type can be narrowband encoded, not every instance of every type can be narrowband encoded due to limitations on sequence sizes inherent in the encoding rules and on message length limitations that may be imposed by implementations.

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 8.4.1.2.3.2.

> 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.

Commentary: 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. The message numbers are defined with each TCIP message definition in Annex C. note that these numbers are not globally unique, but are unique within a business area.

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.

Business Area identifiers are defined as follows:
0-Reserved
1-Scheduling/Blocking/Runcutting (SCH)
2-Passenger Information (PI)
3- Incident Management (IM)
4-Reserved for Testing (RT)
5-Onboard (OB)
6-Control Center (CC)
7- Fare Collection (FC)
8-Spatial Representation (SP)
9-Common Public Transportation (CPT)
10-Transit Signal Priority (TSP/SCP)
11-100-Reserved
101-199 Local Use
200-255 Reserved

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
- Type 6: Choice constructor

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

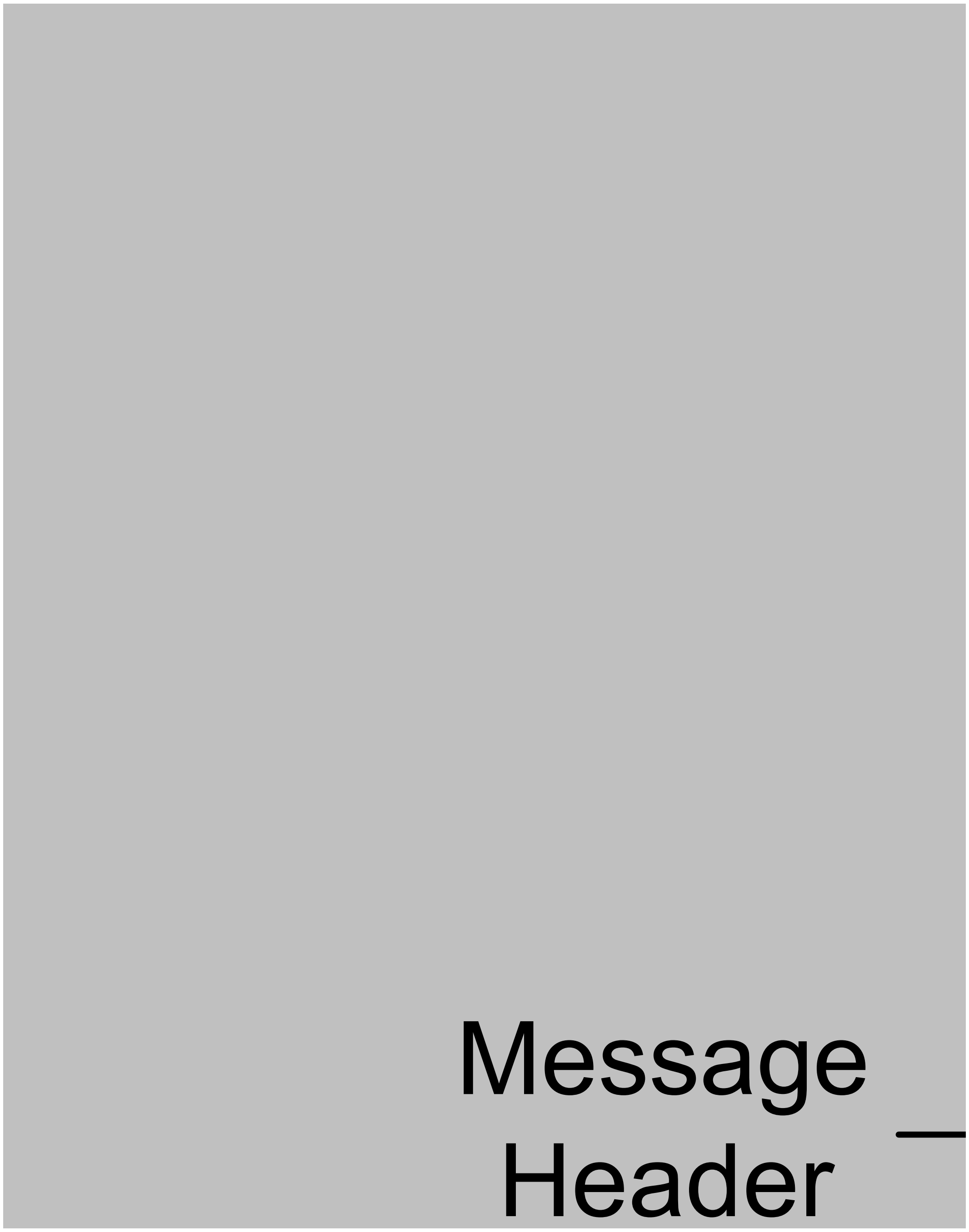


Figure 8.4.1.2.3.1.2: Narrowband TCIP Message Format

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 octet field
BYTE – one octet field
DATE – four octet field
DATETIME – six octet field
DURTIME – four octet field
IDENL – four octet field

IDENS – two octet field
LONG – four octet field
SCHTIME – four octet field
SHORT – two octet field
TIME – four octet field
UBYTE – one octet field
ULONG – four octet field
USHORT – two octet field

The fixed length fields are placed in the message most to least significant byte.

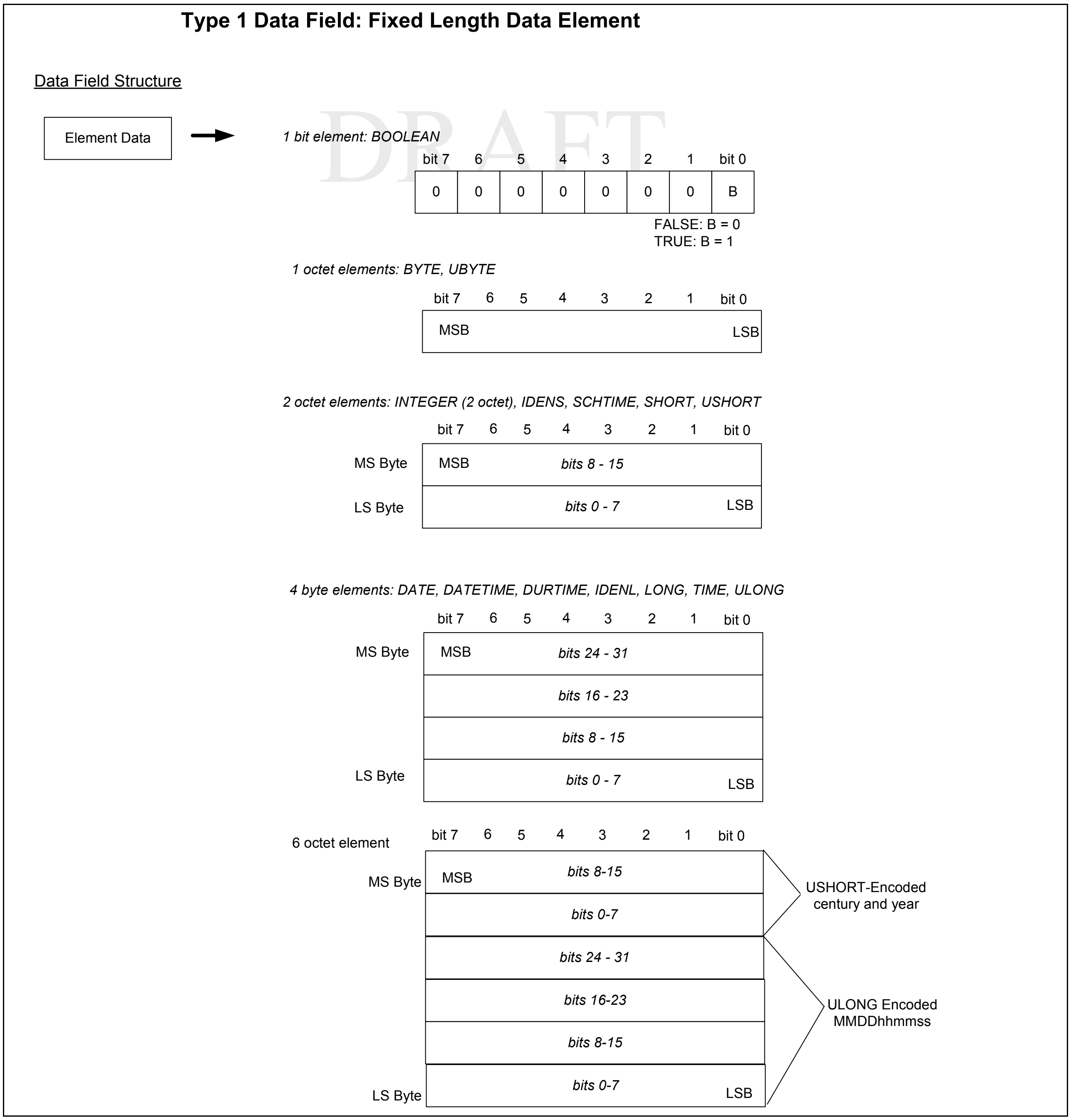


Figure 8.4.1.2.3.2: Narrowband TCIP Fixed Length Element Type

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
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
NAME30	-up to 30 characters
NAME40	-up to 40 characters
NAME60	-up to 60 characters
TELEPHONE	-up to 16 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.

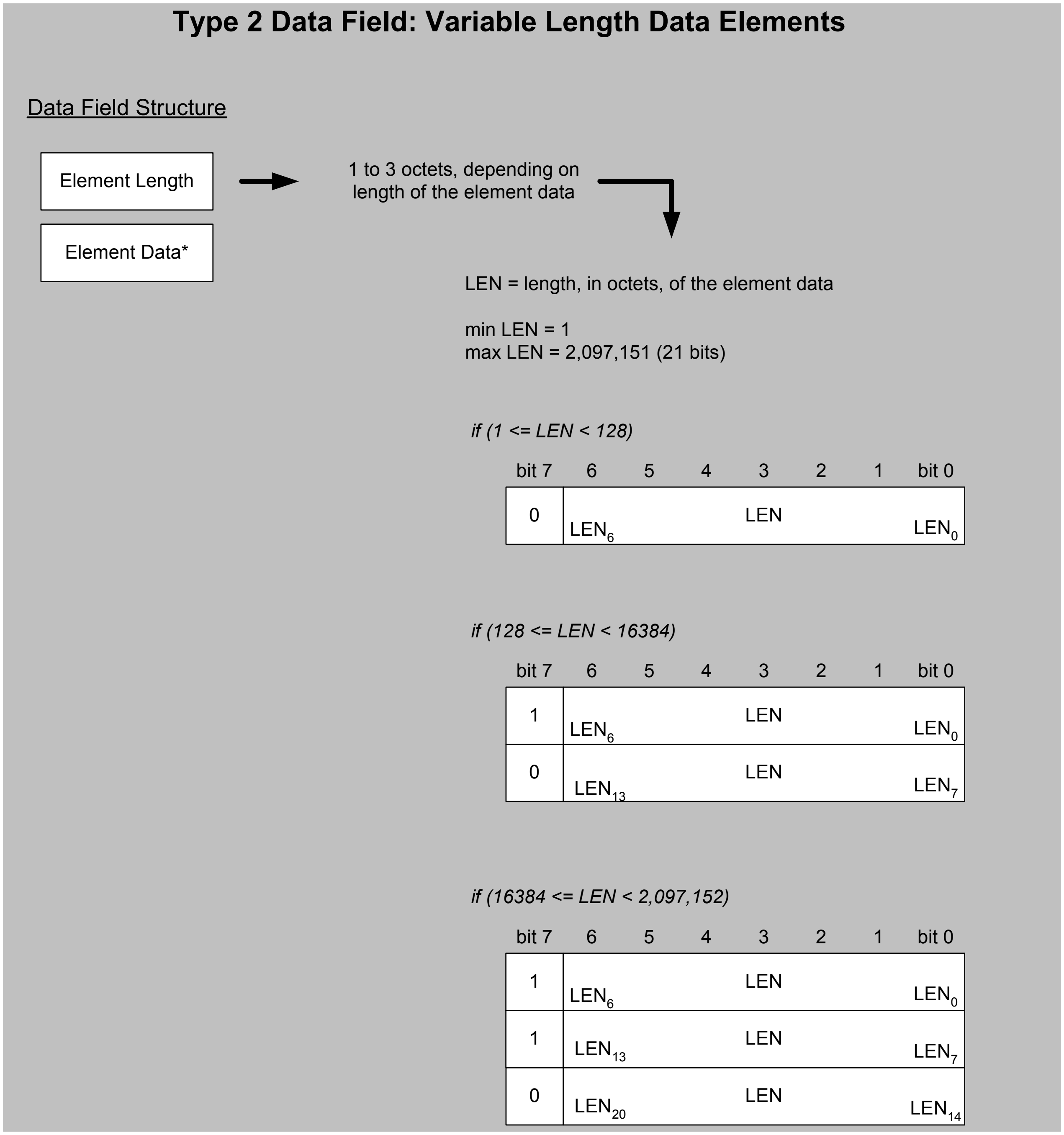


Figure 8.4.1.2.3.3: Narrowband TCIP Variable Length Data Element Type

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.

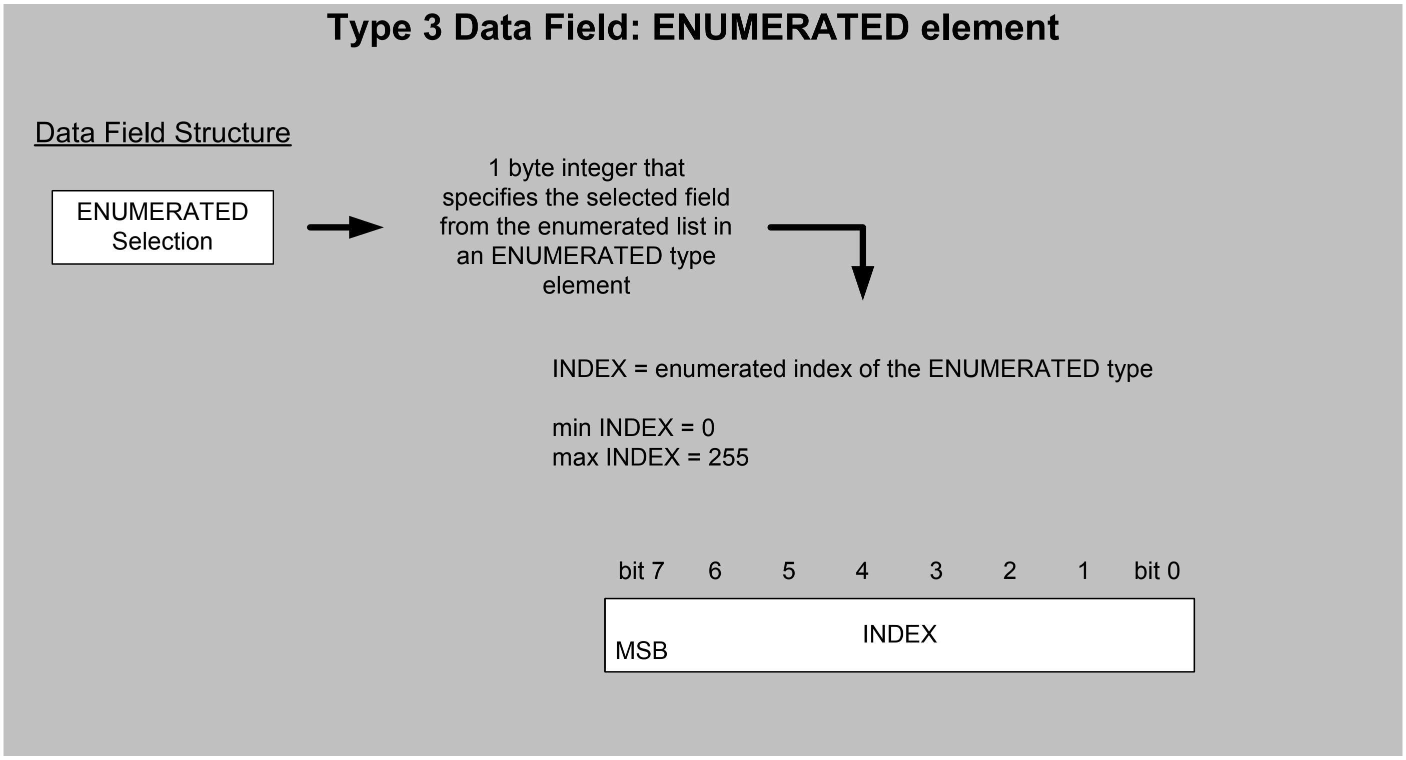


Figure 8.4.1.2.3.4: Narrowband TCIP ENUMERATED Element Type

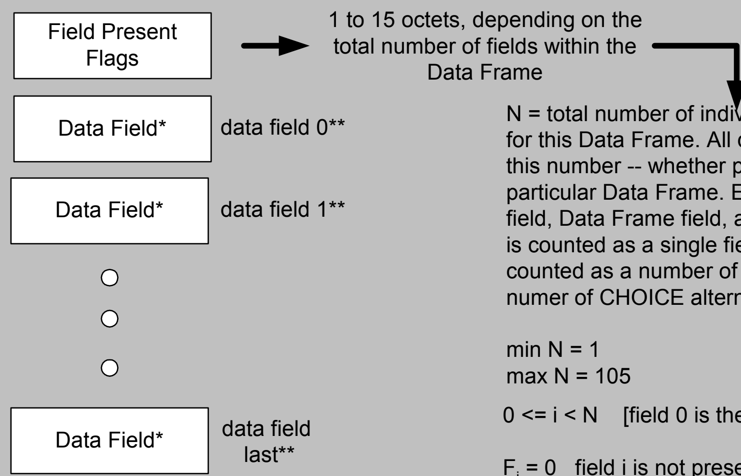
8.4.1.2.3.5 Data Frames

Data frame encoding is performed similarly to the message encoding, but without the message identifier. 1-15 octets 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



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

* 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.

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 octet for each 7 additional fields, setting bit 7 in each header byte to "1", except for the last header octet, 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.

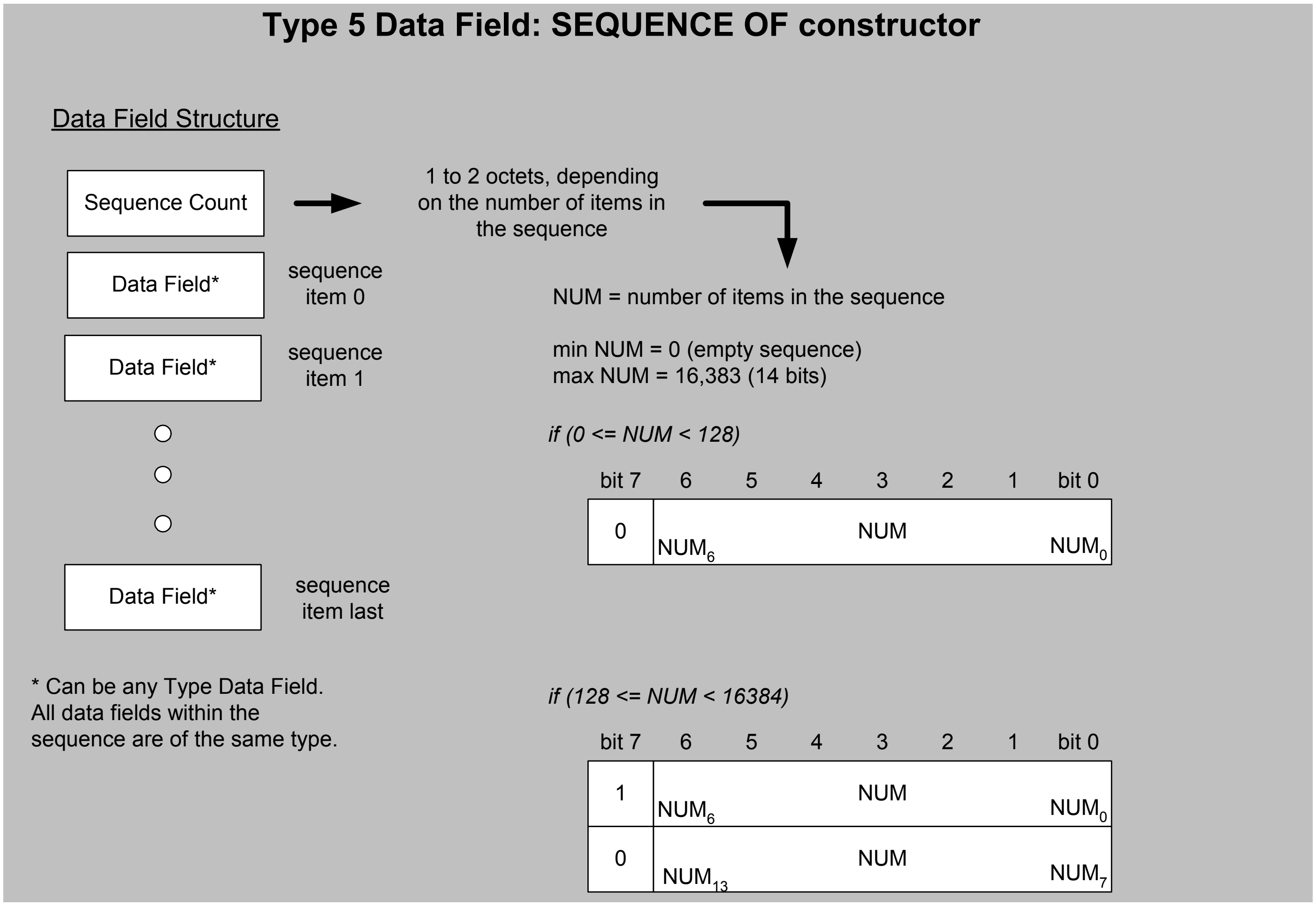


Figure 8.4.1.2.3.6: Narrowband TCIP SEQUENCE OF Type

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.

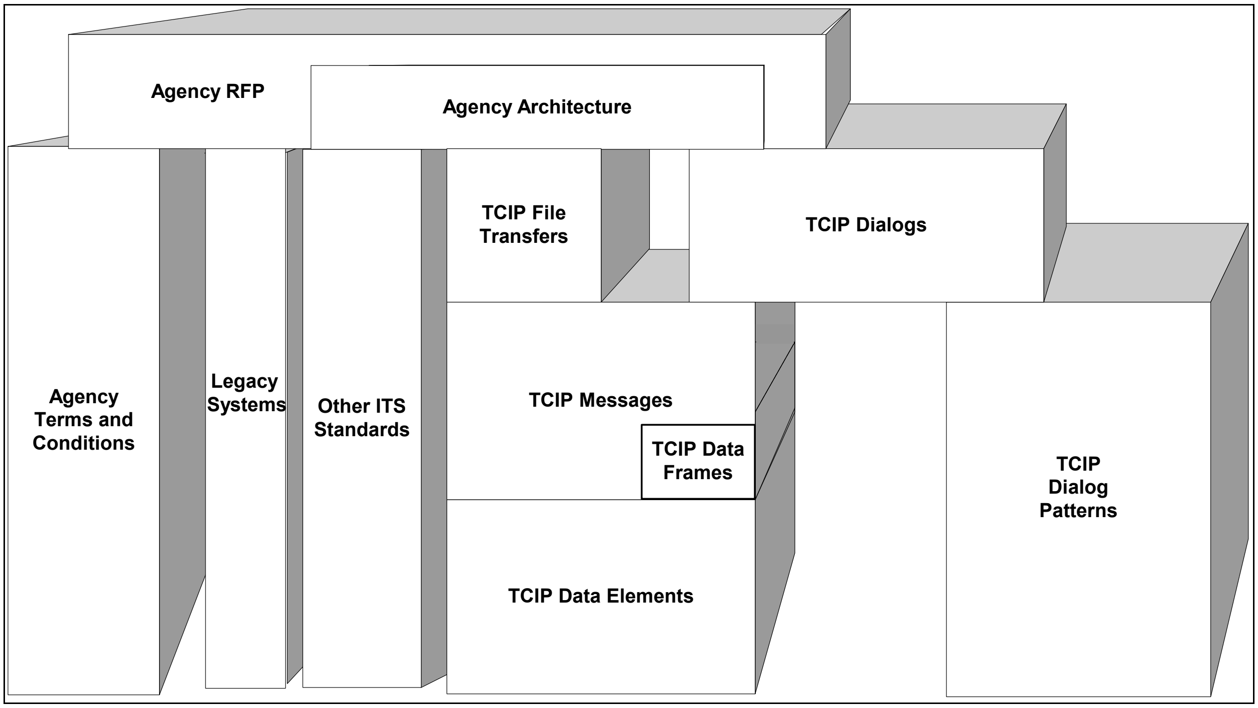


Figure 9.0 TCIP Provides the Building Blocks for Agency Architectures and RFPs

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.

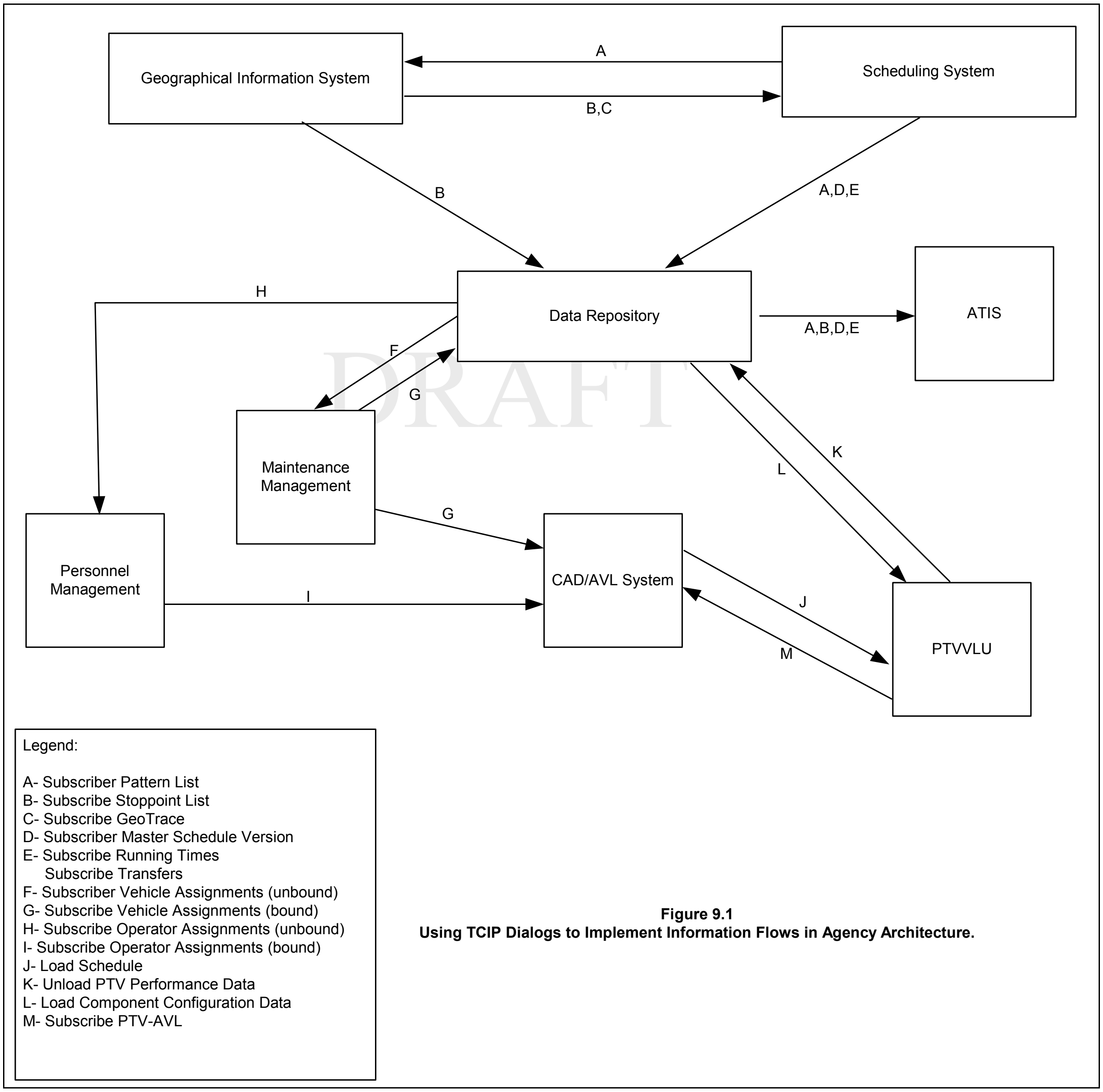


Figure 9.1
Using TCIP Dialogs to Implement Information Flows in Agency Architecture.

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 building blocks for those interfaces (dialogs, file transfers, TCIP Polling).
- Mapping the agency's legacy data and architecture to TCIP data and dialogs.
- Requiring the vendors to include the appropriate TCIP interfaces in their products.
- Specifying what optional and configuration items associated with each interface 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.
- Planning for system test and acceptance including:
  - Planning for PICS delivery and approval in the system design phase of a project
  - Requirement for suppliers to submit TCIP Test Plans and Procedures in the design phase. These documents should, specify how TCIP compliance will be ensured and verified.
  - Requirements for suppliers to demonstrate TCIP compliance at factory acceptance testing.
  - Requirements to verify TCIP compliance of all TCIP interfaces during integration testing and submit test reports and results documenting TCIP conformance.

10 TCIP Polling Protocol

10.1 Purpose of TCIP Polling

The purpose of the TCIP Polling protocol is to provide a standardized TCIP-based protocol for fixed PTV communications over a narrowband agency-owned private radio infrastructure. The protocol must provide for efficient transfer of frequently needed operating information to and from PTVs and must provide a mechanism for conveying TCIP-narrowband-encoded messages and SNMP-encoded messages over the private radio link. This protocol is not intended for use over public/commercial networks, and is not intended to be used for bulk data transfers (loads/unloads) to/from PTVs. The use of this protocol is not required to achieve TCIP or National ITS Architecture conformance within an agency. Agencies may elect to adopt this protocol based on local business needs, or alternatively may elect to adopt existing or vendor-proprietary polling schemes for use with TCIP messages. If an agency adopts the TCIP Polling Protocol as part of its agency architecture, the requirements in Section 10 of this document apply.

10.2 Overview

The polling function provides the following capabilities:

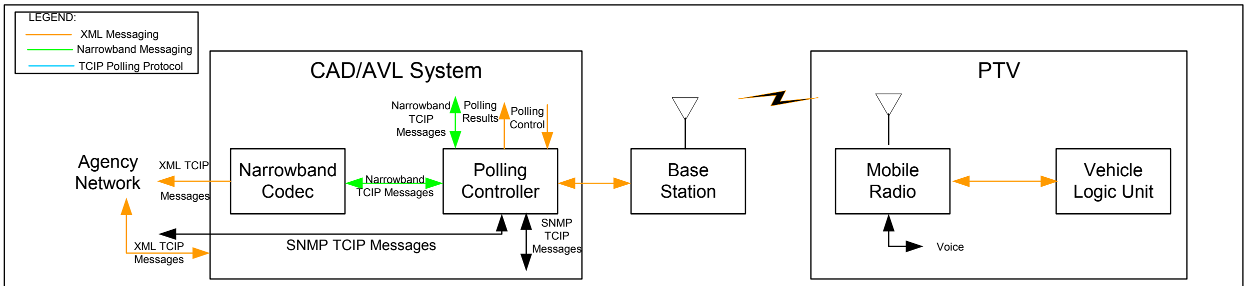
- Allocation/deallocation of polling slots to PTVs
- Unacknowledged transfer of state information from PTVs to the CAD/AVL System via the polling controller
- Priority polls to allow PTVs to send emergency information ahead of the normal poll cycle
- Acknowledged transfer of TCIP messages between PTVs and authorized transit business systems including the CAD/AVL System
- Notification from the polling controller to the CAD/AVL System of vehicles joining, leaving, or losing contact with the polling session
- Fast polling of selected vehicles based on operating conditions
- Group address message transfer to PTVs
- Packetization and reassembly of large messages from a business system to a PTV

The polling controller manages the radio channel and passes messages between the agency's fixed communications network and the PTV (VLU). PTV operating status information is passed to the polling controller in poll responses from the PTV. The polling controller forwards this status information to the CAD/AVL System using the "Notify PTV Polling Result" dialog. The polling controller by default requests the minimum operating information with each poll. The polling controller may be configured by the CAD/AVL System to request additional information on a PTV-specific basis using the Subscribe PTV-Polled Parameters dialog.

The polling controller may send a narrowband TCIP message an SNMP TCIP message, or a packet of a long TCIP narrowband-encoded message (in a message wrapper) to any PTV in the polling session at any time, except when that PTV is the 'currently polled' PTV. The PTV may send a narrowband-encoded TCIP message or an SNMP-encoded TCIP message to the polling controller immediately following a Poll Response. The polling function defines the interactions between a "polling controller" and a PTV. The functions described in this section can be implemented in a variety of ways as shown in the following scenarios.

10.2.1 Scenario 1

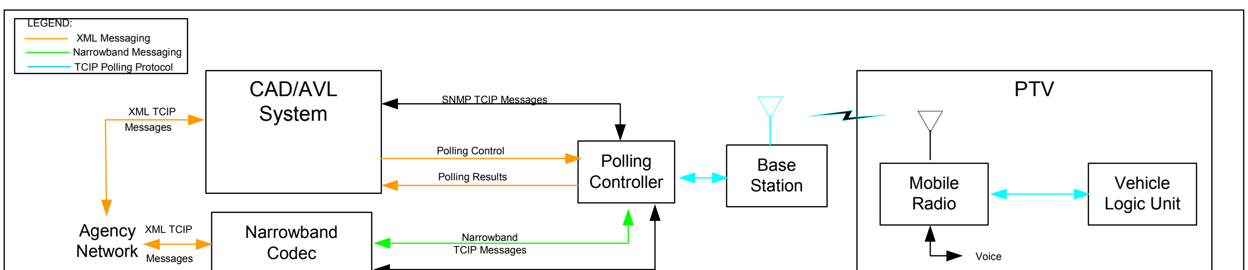
This scenario incorporates the Polling Controller with the CAD/AVL System. Figure 10.2.1 depicts the relationships in this scenario. The narrowband codec provides a two-way translation function between an agency-specified list of TCIP-XML messages and their TCIP narrowband encoded equivalents.



Scenario #1
Figure 10.2.1

10.2.2 Scenario 2 Separate Units

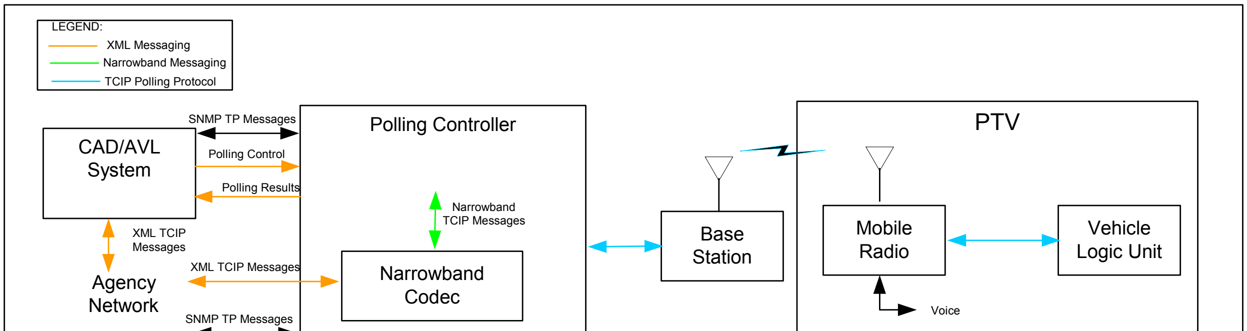
This scenario separates the polling controller, CAD/AVL System, and Narrowband Codec into separate physical devices. This approach may be useful where narrowband messages are to be distributed to more than one polling controller, or to a non-TCIP polling controller. TCIP does not specify the wrapper or protocol to be used to convey narrowband messages between the Narrowband Codec and the polling controller. Figure 10.2.2 depicts the relationships in this scenario.



Scenario #2
Figure 10.2.2

10.2.3 Scenario 3 Integrated Codec/Polling controller

This scenario incorporates the Narrowband Codec into the polling controller. Figure 10.2.2 depicts the relationships in this scenario.



Scenario #3
Figure 10.2.3

10.3 Assumptions

The TCIP Polling Protocol is based on the following assumptions:

- The agency owns a private radio network, that is shared for voice and data. Separate channels are allocated for voice and data. The Operator Initiated Voice Radio Call and Dispatcher Initiated Voice Radio Call dialogs are used to control the transfers between voice and data modes.
- Radio system characteristics including band, channel allocation, bit rate and radio performance vary from agency to agency. See section 10.8 for a list of parameters that affect the operation of the TCIP Polling Protocol.
- Base (fixed) radios communicate on the voice and data channel(s) simultaneously, and possibly in full duplex mode on the data channel. Mobile radios communicate on only one channel (voice or data) at a time and may be limited to simplex two-way alternate mode on the data channel.
- All PTVs (on power-up) are initially not allocated to a polling slot. A PTV which has been turned off or out of contact must 'join' the polling session and be allocated a slot by the polling controller.
- The protocol allows for very large numbers of PTVs on a channel as well as for very long message lengths (approximately 64k in each case) however, real-world implementations will be limited to much smaller values in each case to achieve reasonable performance levels. In particular agencies will assign upper limits on narrowband-encoded TCIP message length. These maximum message lengths may be different for messages to and from the PTV.

10.4 Definitions

Allocation Update – a data structure sent across the radio channel from the polling controller to the PTV announcing recent allocations and deallocations.

Polling controller – A fixed software entity that controls access to a data channel, by allocating and deallocating polling slot numbers to mobile units, and by issuing polls that allow PTVs to access the channel.

Join Request – A Data structure sent across the radio channel from the PTV to the polling controller requesting that the PTV be allocated a polling slot number. Also used in response to a priority poll to notify the polling controller of an urgent message queued on the PTV.

Leave Request – a data structure sent across the radio channel from the PTV to the polling controller requesting deallocation of its polling slot.

Message Wrapper – A data structure sent across the radio channel in either direction conveying a TCIP – Narrowband –encoded message.

Poll – A data structure sent across the radio channel from the polling controller to the PTV giving the PTV permission to transmit.

Poll Response – A data structure, containing PTV operating information, sent across the radio channel from the PTV to the polling controller in response to a poll.

Priority Poll – A data structure sent across the radio channel from the polling controller to the PTV to allow any PTV to notify the polling controller of an urgent message waiting.

Session Poll – a data structure sent from the polling controller to the PTV to allow a PTV without a polling slot to send a join request.

10.5 Data Structure Definitions

The data structures used by the TCIP Polling Protocol are the: Poll, Poll Response, Priority Poll, Session Poll, Packeted Message Wrapper, Narrowband Message Wrapper, SNMP Message Wrapper, Group Reset, Join Request, Leave Request, and Allocation Update. All of these structures share the same basic data frame structure. Each data structure is defined in the following subsections.

10.5.1 Basic Polling Data Frame Structure

The Basic Polling Data Frame structure is used to convey all TCIP Polling Protocol Data Structures. It is defined as follows:

FIELD #	NAME	CONTENT	NOTES
1	Bit Sync	Minimum 1 octet value AAH	Agencies/vendors may extend the length of this field by adding additional octets of AAH if the radio system cannot reliably achieve bit sync in one octet time.
2	Frame Sync Flag	One octet value 7EH	Allows the receiver to detect the end of bit sync and transition to start of contents.
3	Slot Number	Two octet field-most significant octet first	Identifies the slot number associated with this data structure.
4	Length	Two octet field denoting the length in octets of the contents of this structure	Measured from the Frame Sync Flag to the End Sync Flag – non-inclusive.
5	ID	Octet field identifying the type of this data structure.	Values 01-10H are reserved for TCIP Business areas, and signify a message wrapper. Other values are identified with each data structure.
6-N	Structure Type-specific content		
N+1	Checksum	Modulo – 256 sum of values in fields 3-N	
N+2	End Sync Flag	One octet value 7EH	Allows receiver to verify end of transmission

Important Note: the decoders in both the TCIP Controller and the VLU frame by calculating the octets left until the end of the frame. The End Sync Flag shall be used only as a verification that the receiver remained in sync throughout the frame. Nonconformant use of the end sync flag as a “trigger” will result in random premature end of frame detections and failures to decode frames.

10.5.2 Poll Data Structure

The Poll Data Structure is used to elicit a response from a PTV, after the PTV has been allocated a polling slot. The Poll Contents field contains fields to acknowledge the most recently received message wrapper from the PTV and to direct the PTV as to what information should be returned in the poll response.

FIELD #	NAME	CONTENT	NOTES
1	Bit Sync	Minimum 1 octet value AAH	Agencies/vendors may extend the length of this field by adding additional octets of AAH if the radio system cannot reliably achieve bit sync in one octet time.
2	Frame Sync Flag	One octet value 7EH	Allows the receiver to detect the end of bit sync and transition to start of contents.
3	Slot Number	Two octet field-most significant octet first	Identifies the mobile, by slot number, that should respond to this poll.
4	Length	Two octets-variable	
5	ID	1 octet field value =A1H	
6	Poll Contents	Narrow-band encoded TCIP data frame.	CCPollContents
7	Checksum	Modulo – 256 sum of values in fields 3-N	
8	End Sync Flag	One octet value 7EH	Allows receiver to verify end of transmission

10.5.3 Poll Response Data Structure

The Poll Response Data Structure provides the PTV response to a poll from the PTV to the polling controller. This structure has three alternative identifiers:

- A2H indicates the poll response is the complete data transmission from the PTV to the polling controller.
- A3H indicates that the poll response is followed by a message wrapper from the PTV to the polling controller.

Note that field #6 uses a TCIP data frame, not a message as its contents.

FIELD #	NAME	CONTENT	NOTES
1	Bit Sync	Minimum 1 octet value AAH	Agencies/vendors may extend the length of this field by adding additional octets of AAH if the radio system cannot reliably achieve bit sync in one octet time.
2	Frame Sync Flag	One octet value 7EH	Allows the receiver to detect the end of bit sync and transition to start of contents.
3	Slot Number	Two octet field-most significant octet first	Identifies the mobile, by slot number, sending the poll response
4	Length	Two octets-variable	
5	ID	A2H – No wrapper A3H – Wrapper follows	
6	Response Contents	Narrow-band encoded TCIP –data frame	CCPollResponseContents
7	Checksum	Modulo – 256 sum of values in fields 3-N	
8	End Sync Flag	One octet value 7EH	Allows receiver to verify end of transmission

10.5.4 Priority Poll

The Priority Poll Data Structure is used by the polling controller to create a contention-based opportunity for a PTV to send a Join Request signifying a high-priority message to the CAD/AVL System without waiting for its regular poll. The selection of what messages the Vehicle Logic Unit deems high-priority is local agency-defined.

FIELD #	NAME	CONTENT	NOTES
1	Bit Sync	AAH	Agencies/vendors may extend the length of this field by adding additional octets of AAH if the radio system cannot reliably achieve bit sync in one octet time.
2	Frame Sync Flag	7EH	Allows the receiver to detect the end of bit sync and transition to start of contents.
3	Slot Number	FFFFH	Broadcast- for use by any authorized PTV
4	Length	Two octets 0006H	
5	ID	One octet A4H	
6	Checksum	Modulo – 256 sum of values in fields 3-N	
7	End Sync Flag	One octet value 7EH	Allows receiver to verify end of transmission

10.5.5 Session Poll

The Session Poll Data Structure is used by the polling controller to invite PTVs that do not have an allocated polling slot to request a slot.

FIELD #	NAME	CONTENT	NOTES
1	Bit Sync	AAH	
2	Frame Sync Flag	7EH	Allows the receiver to detect the end of bit sync and transition to start of contents.
3	Slot Number	FFFFH	Broadcast- for use by any unallocated PTV
4	Length	Two octets 0006H	
5	ID	One octet A5H	
6	Checksum	Modulo – 256 sum of values in fields 3-N	
7	End Sync Flag	One octet value 7EH	Allows receiver to verify end of transmission

10.5.6 Narrowband Message Wrapper

The Narrowband Message Wrapper Data Structure is used by the PTV or the polling controller to convey a Narrowband TCIP message.

FIELD #	NAME	CONTENT	NOTES
1	Bit Sync	AAH	
2	Frame Sync Flag	7EH	Allows the receiver to detect the end of bit sync and transition to start of contents
3	Slot Number		Identifies the mobile originator or destination to which the message pertains. Value FFFFH indicates all mobiles (broadcast from polling controller)
4	Length	Two octets 0005H	Measured from the Frame Sync Flag to the End Sync Flag – non-inclusive.
5	ID	Identifies the TCIP business area	Value 01-20H. This matches as the BID field for the narrowband encoded message.
6	Fixed IP address	16 octets	IPV6 Format, Same as CPT-IP-Address
7	Fixed Port	1 octet	Same as CPT-UDP-TCP-Port-Number
8	Message Number	1 octet	Same as CC-MessageCounter
9	Last Received Message Number	1 octet	Same as CC-MessageCounter
10	Message Data	TCIP Narrowband-Encoded Message	Variable Length
11	Checksum	Modulo – 256 sum of values in fields 3-10	
12	End Sync Flag	One octet value 7EH	Allows receiver to verify end of transmission

10.5.7 Join Request Data Structure

The Join Request Data Structure is used by the PTV to request to join the TCIP polling session, or in response to a Priority Poll to notify the polling controller of a high-priority message.

FIELD #	NAME	CONTENT	NOTES
1	Bit Sync	Minimum 1 octet value AAH	
2	Frame Sync Flag	One octet value 7EH	Allows the receiver to detect the end of bit sync and transition to start of contents.
3	Slot Number	Two octets	0000H in response to a Session Poll, assigned slot number in response to a priority poll.
4	Length	Two octets – 000AH	
5	ID	One octet – Value = A6H	
6	vehicle ID	Four octets	Same value as CPT-VehicleID to identity requester
7	Checksum	Modulo – 256 sum of values in fields 3-N	
8	End Sync Flag	One octet value 7EH	Allows receiver to verify end of transmission

10.5.8 Leave Request Data Structure

The Leave Request Data Structure is used by a PTV to ask to be removed from the polling cycle by the polling controller. The Leave Request is sent in lieu of a Poll Response by the PTV.

FIELD #	NAME	CONTENT	NOTES
1	Bit Sync	Minimum 1 octet value AAH	
2	Frame Sync Flag	One octet value 7EH	Allows the receiver to detect the end of bit sync and transition to start of contents.
3	Slot Number	Two octets	Currently assigned slot
4	Length	Two octets – Value = 000AH	
5	ID	One octet – Value = A7H	
6	Vehicle ID	Four octets	Same value as CPT-VehicleID
7	Checksum	Modulo – 256 sum of values in fields 3-N	
8	End Sync Flag	One octet value 7EH	Allows receiver to verify end of transmission

10.5.9 Allocation Update Data Structure

The Allocation Update Data Structure is used by the polling controller to notify PTVs that they have been added or deleted from the poll cycle. A history of the last 6 adds and deletes are included to allow multiple opportunities for a PTV to receive a copy of the Allocation Update and recognize its revised status.

The polling controller shall not include the same slot identifier in the added and deleted lists except for slot zero which is always valid.

FIELD #	NAME	CONTENT	NOTES
1	Bit Sync	Minimum 1 octet value AAH	
2	Frame Sync Flag	One octet value 7EH	Allows the receiver to detect the end of bit sync and transition to start of contents.
3	Slot Number	FFFFH	Broadcast
4	Length	Two octets – Value =0037H	
5	ID	One octet – Value = A8	
6	Delete all	Boolean (1 octet)	Value 1 all slots rescinded due to polling controller restart, value 0 normal operations.
7	Added slot 1	Two Octets	Slot # Assigned
8	Added Vehicle 1	Four Octets	CPT-VehicleID
9	Added slot 2	Two Octets	Slot # Assigned
10	Added Vehicle 2	Four Octets	CPT-VehicleID
11	Added slot 3	Two Octets	Slot # Assigned
12	Added Vehicle 3	Four Octets	CPT-VehicleID
13	Added slot 4	Two Octets	Slot # Assigned
14	Added Vehicle 4	Four Octets	CPT-VehicleID
15	Added slot 5	Two Octets	Slot # Assigned
16	Added Vehicle 5	Four Octets	CPT-VehicleID
17	Added slot 6	Two Octets	Slot # Assigned
18	Added Vehicle 6	Four Octets	CPT-VehicleID
19	Deleted slot 1	Two Octets	Slot # Vacated
20	Deleted slot 2	Two Octets	Slot # Vacated
21	Deleted slot 3	Two Octets	Slot # Vacated
22	Deleted slot 4	Two Octets	Slot # Vacated
23	Deleted slot 5	Two Octets	Slot # Vacated
24	Deleted slot 6	Two Octets	Slot # Vacated
25	Checksum	Modulo – 256 sum of values in fields 3-N	
26	End Sync Flag	One octet value 7EH	Allows receiver to verify end of transmission

10.5.10 Packetized Message Wrapper

The Packetized Message Wrapper is used by the Polling Controller to convey a segment of a long TCIP narrowband message to a PTV.

Field #	Name	Content	Notes
1	Bit Sync	AAH	Agencies/vendors may extend the length of this field by adding additional octets of AAH if the radio system cannot reliably achieve bit sync in one octet time.
2	Frame Sync Flag	7EH	Allows the receiver to detect the end of bit sync and transition to start of contents.
3	Slot Number		Identifies the mobile(s) to which this frame is sent.
4	Length	Two Octets-variable	Measured from the Frame Sync Flag to the End Sync Flag – non-inclusive.
5ID	Two Octets-identifies the TCIP Business Area	Value = BI-DHO. This matches the BID field plus BOH for the narrowband-encoded message	
6	Fixed IP Address	16 Octets	IPV6 Format, Same as CPT-IP-Address
7	Fixed Port	1 octet	Same as CPT-UDP-TCP-Port-Number
8	Message Number	1 octet	Same as CC-MessageCounter
9	Last Received Message Number	1 octet	Same as CC-MessageCounter
10	Segment Numbers	1 octet	High order 4 bits = m,low order 4 bits=n. This is segment m of n.
11	Message Data	TCIP-Narrowband Encoded Message Segment	Variable Length
12	Checksum	Modulo – 256 sum of fields 3-11	
13	End Sync Flag	1 octet Value 7EH	Allows receiver to verify end of transmission

10.5.11 SNMP Message Wrapper

The SNMP Message Wrapper Data Structure is sued to convey SNMP-encoded messages over the TCIP-Polled radio link.

Field #	Name	Content	Notes
1	Bit Sync	Minimum 1 octet value = AAH	Agencies/vendors may extend the length of this field by adding additional octets of AAH if the radio system cannot reliably achieve bit sync in one octet time.
2	Frame Sync Flag	1 Octet value 7EH	Allows the receiver to detect the end of bit sync and transition to start of contents.
3	Slot Number	2 Octet field, most significant octet first	Identifies the mobile that originated, or should receive this frame.
4	Length	2 Octet field-variable	
5	ID	The octet value = AAH	
6	Fixed IP Address	16 Octets	IPV 6 Format – Same as CPT-IP-Address
7	Fixed Port	1 Octet	Same as CPT-UDP-TCP-Port Number
8	Message Number	1 Octet	Same as CC-MessageCounter
9	Last Received Message	1 Octet	Same as CC-MessageCounter
10	Message Data	TCIP-SNMP Encoded Message	Variable Length
11	Checksum	Modulo-256 sum of values in fields 3-10	
12	End Sync Flag	One Octet Value 7EH	Allows receiver to verify end of transmission

10.5.12 Group Reset

The Group Reset Data Structure is used to notify PTVs that one or more polling group has been reset. If not enough groups are reset to fill the frame, the last reset group is repeated until the frame is filled.

Field #	Name	Content	Notes
1	Bit Sync	Minimum 1 octet value AAH	Agencies/vendors may extend the length of this field by adding additional octets of AAH if the radio system cannot reliably achieve bit sync in one octet time.
2	Frame Sync Flag	7EH	Allows the receiver to detect the end of bit sync and transition to start of contents.
3	Slot Number	FFFFH	Broadcast-let everybody know about this event
4	Length	2 Octets value = 0010H	
5	1D	1 Octet value = A9	
6	Reset Group 1	Number of Group to Reset	Type = CC=PollingGroup
7	Reset Group 2	Number of Group to Reset	Type = CC=PollingGroup
8	Reset Group 3	Number of Group to Reset	Type = CC=PollingGroup
9	Reset Group 4	Number of Group to Reset	Type = CC=PollingGroup
10	Reset Group 5	Number of Group to Reset	Type = CC=PollingGroup
11	Reset Group 6	Number of Group to Reset	Type = CC=PollingGroup
12	Reset Group 7	Number of Group to Reset	Type = CC=PollingGroup
13	Reset Group 8	Number of Group to Reset	Type = CC=PollingGroup
14	Reset Group 9	Number of Group to Reset	Type = CC=PollingGroup
15	Reset Group 10	Number of Group to Reset	Type = CC=PollingGroup
16	Checksum	Modulo 256 sum of values in fields 3-15	
17	End Sync Flag	1 Octet value = 7EH	Allows receiver to verify end of transmission.

10.6 Polling Controller Rules of Procedure

10.6.1 Allocation of Slot Numbers

The polling controller shall allocate slot numbers in the range 0101H – FFFEH. Slot number 0000H shall be reserved as a null slot number, and slot FFFFM shall be reserved as a broadcast slot number. The polling controller shall assign slot numbers beginning with 0101H and increasing sequentially, skipping any already assigned slot numbers. Upon allocating slot number FFFEH, the polling controller shall make new allocations starting with slot number 0101H, or the lowest currently unallocated slot number. Slot numbers 0001H – 00FFH shall be reserved for PTV group addresses.

The polling controller shall allocate slot numbers only to PTVs requesting a slot allocation by sending a Join Request to the polling controller. When the polling controller allocates a slot number, it shall immediately send an Allocation Update. The polling controller may repeat the Allocation Update transmission $N_{ALLOCTRETRY}$ times, to increase the probability of success. The polling controller shall log each allocation event.

For each allocated slot, the polling controller shall maintain the following slot data:

- Slot-number (from allocation algorithm)
- PTV ID (from join request)
- Last-message-number-sent (initialize to zero)
- Last-message-number-received (initialize to zero)
- Queue of up to $N_{CTLPTVQ}$ (default 8) TCIP messages to the PTV (initialize to empty). Elements in this queue may be SNMP-encoded messages in a SNMP message wrapper, TCIP-narrowband encoded messages in a narrowband message wrapper or a segment of a TCIP narrowband –encoded message in a Packetized Message Wrapper
- Message-try-count (initialize to zero)
- Message ready to send (initialize to false)
- Poll-no-response-count (initialize to zero)
- Agency-data (CC-AgencyData=Initialize to none)
- Agency-data-try-count (initialize to zero)
- Poll-data (initialize to zero)

10.6.2 Deallocation of Slot Numbers

The polling controller shall deallocate slot numbers upon incrementing the Poll-no-response-count for the slot to (1+ $N_{MAXBADPOLL}$), or upon receipt of a Leave Request from the PTV.

10.6.3 Session Polls

The polling controller shall issue a Session Poll to the radio channel at least once per polling cycle, and at least once per $T_{SESSIONPOLL}$ seconds. Upon issuing a Session Poll the polling controller shall refrain from issuing any additional polls (of any kind) while it waits for a poll answer.

10.6.4 Priority Polls

The polling controller shall issue a Priority Poll to the radio channel at least once per polling cycle or at least once per $T_{PRIORITYPOLL}$ seconds. Upon issuing a Priority Poll, the polling controller shall refrain from issuing any additional polls (of any kind) while it waits for a poll answer.

10.6.5 Normal Polls

The polling controller shall issue Polls to the radio channel on a cyclical basis to each allocated slot number. Additionally up to $N_{MAXFASTPOLL}$ PTVs may be assigned to be 'fast polled' at a rate of once every $T_{FASTPOLLINTERVAL}$ seconds. Fast polls are identical to other polls, except the designated PTVs are polled every $T_{FASTPOLLINTERVAL}$ in addition to their normal poll cycle. The polling controller may be optionally configured to automatically provide additional polls to vehicles with Poll Responses reporting queued messages to the polling controller.

When the polling controller issues a poll to a PTV, it shall populate the poll data structure using the internally stored values for the PTV to be polled including Last-message-number-received, Poll-data, and Agency-data (if present). If Agency data is present, the polling controller shall increment the Agency-data-try-count and shall compare it with the configured value $N_{AGENCYDATAMAXTRIES}$. If the Agency-data-try-count is greater or equal to $N_{AGENCYDATAMAXTRIES}$, then the polling controller shall delete the agency data and shall reset Agency-data-try-count to zero.

After issuing a poll, the polling controller shall refrain from issuing any additional polls (of any kind) while it waits for a poll answer.

10.6.6 Waiting for a Poll Answer

Upon issuing a Session or Priority Poll, the polling controller shall wait for a valid received answer (Join Request) for up to $T_{SESSIONWAIT}$ milliseconds. If a valid Join Request is received, the polling controller shall cease waiting and immediately process it as described in section 10.6.8. If a Join Request is not received and $T_{SESSIONWAIT}$ expires, the polling controller shall resume polling.

Upon issuing a Normal Poll, the polling controller shall wait for a valid received answer (Poll Response) for up to $T_{NORMALWAIT}$ milliseconds. If a valid Poll Response is received, the polling controller shall cease waiting and immediately process it as described in section 10.6.11. If a Poll Response is not received and $T_{NORMALWAIT}$ expires, the polling controller shall:

- Increment the Poll-no-response-count variable for the polled PTV
- If Poll-no-response-count >= $N_{MAXBADPOLL}$ then
  - Log a loss of contact event
  - Deallocate the polling slot for the polled PTV
- Resume polling (with next allocated slot number)

10.6.7 Sending a PTV Message

The polling controller shall send messages to PTVs interspersed with polls "messages" in this context include:

- A complete narrowband TCIP message in a message wrapper
- A complete SNMP encoded TCIP message
- A segment of a TCIP-narrowband encoded message in a packetized message wrapper

If the base station(s) is full-duplex, the polling controller may send a message to a PTV while waiting for an answer to a poll, however the polling controller shall not send a message to a PTV while it is waiting for an answer to a poll from that specific PTV. The polling controller shall send each pending outbound message at least once per polling cycle, until it is acknowledged by a received SNMP or Narrowband message wrapper, or Poll Response, or discarded based on the value of PTV Message Try Count (see below).

The polling controller shall increment the PTV message try count each time the PTV is polled. If the Message-try-count exceeds $N_{MSGMAXTRIES}$, the polling controller shall:

- Reset Message-try-count to zero for that PTV
- Discard the message at the head of the queue for that PTV
- If there is another message in the queue to that PTV, assign the next sequential message number to the new message at the head of the queue.

10.6.8 Receiving a Join Request

Upon receipt of a Join Request in response to a Session Poll, the polling controller shall allocate a polling slot to the requesting PTV (see 10.6.1), and check for any duplicate slot for the same PTV. If the polling controller finds a duplicate polling slot assignment, the polling controller shall move the message queue, poll data and agency data from the old slot to the new slot and deallocate the old slot. The polling controller shall send an Allocation Update immediately, or immediately following the current message or poll transmission if one is in progress. The Allocation update shall be transmitted $N_{ALLOCRETRY}$ times, after which the polling controller shall resume the poll cycle.

10.6.9 Receiving a Message Wrapper

Upon receipt of a message wrapper from the PTV, the polling controller shall:

- Check the last received message number in the wrapper and compare with the message to the PTV awaiting acknowledgement (if any). If the message is acknowledged:
  - Reset PTV Message Try Count to zero for that PTV
  - Discard the message at the head of the queue for that PTV
  - If there is another message in the queue to that PTV, assign the next sequential message number to the message at the head of the queue.
- Check the message number in the wrapper and determine if it matches Last-message-number-received for that PTV.
  - If the numbers match discard the message wrapper (duplicate)
  - If the numbers don't match set Last-message-number-received to the value in the wrapper and forward the message to its recipient.

10.6.10 Receiving a Leave Request

Upon receipt of a Leave Request, the polling controller shall deallocate the polling slot of the requesting PTV, discard any queued message for that PTV, and send an Allocation Update immediately or immediately following the current message or poll transmission if one is in progress.

10.6.11 Receiving a Poll Response

Upon receipt of a Poll Response from a PTV, the polling controller shall:

- Check the last received message number in the response and compare with the message to the PTV awaiting acknowledgement (if any). If the message is acknowledged:
  - Reset PTV Message Try Count to zero for that PTV
  - Discard the message at the head of the queue for that PTV
  - If there is another message in the queue to that PTV, assign the next sequential message number to the message at the head of the queue.
- Package the CcPollResponseContents for the Poll Response into a CcPTVPollInfo message and send to the CAD/AVL System (Notify PTV Polling Result dialog.)

10.6.12 Receiving a CcPollParameters Message

The polling controller shall establish and maintain an event-driven subscription to polling control and configuration information from the CAD/AVL System using the Subscribe Poll Control dialog. Upon receipt of a CcPollParameters message from the CAD/AVL System, the polling controller shall:

- Update any configured timer and/or counter values to the values specified in the received message.
- Set the fast poll PTV list to match the list in the received message, or remove all PTVs from the list if no fast poll PTVs are listed in the message.
- If any init-polling-groups are present:

  - For each init-polling-group present find all stored PTV records with the stored group ID set to the received group-id value and update the stored GroupID field to zero. Commentary: Newly initialized groups should have no members.
  - For each init-polling-group present create a group record for the initialized group containing the group id, the group's IP address, and a last-message-sent counter (initialize to zero) for that group. Overwrite or delete any old group record containing the same group ID.
  - For each init-polling-group log the initialization event
  - Send as many Group Reset frames as are required to reset all groups specified by the CcPollParameters message
- If any add-group-PTVs are present; find the indicated PTV's stored data, and update the Group ID field. If a PTV is listed, but does not have an allocation, log this event.
- For each PTV for which a PTV-poll-infosets entry is present:
  - If a poll-data field is present, replace the Poll-data for that PTV with the received value.
  - If an agency-data field is present, replace the stored Agency-data for that PTV with the received value and set the Agency-data-try-count to zero.
  - Update the stored Group-id to the received value provided in group-id

10.6.13 Receiving a TCIP Message for Delivery to a PTV

The polling controller upon receiving a message from an agency for transmission to a PTV shall:

1. Determine if the message length exceeds $N_{MASMSGLENTOPTV}$ for a narrowband encoded message or exceeds $N_{MAXPACKET}$ for an SNMP encoded message.
2. Determine if the message is to a PTV with an allocated slot, if the PTV's slot is not allocated, the polling controller shall discard the message and log the event
3. Determine if the message is a narrowband encoded message of length less than $N_{MAXPACKET}$, a narrowband encoded message of length greater than $N_{MAXPACKET}$, or an SNMP encoded message.
4. If the message is a narrowband encoded message of length less than $N_{MAXPACKET}$:
   a. Encode the message in a narrowband message wrapper
   b. Determine if the addition of the message to the queue for the destination PTV would cause the queue length to exceed $N_{CTLPTVQ}$, and if so discard the message an log the event
   c. Otherwise add the message to the queue for the PATV. If the message is at the head of the queue, transmit the message and set the Message-try-count to zero.
5. If the Message is a narrowband encoded message of length greater than $N_{MAXPACKET}$ and less than or equal to $N_{MAXNSGLEN}$:
   a. Encode the message into a sequence of packetized message wrappers. Include $N_{MAXPACKET}$ octets of the message into each wrapper until the final wrapper has less than $N_{MAXPACKET}$ octets to include.
   b. Determine if the addition of the message segments to the queue for the destination PTV would cause the queue length to exceed $N_{CTLPTVQ}$, and if so discard the message (all segments) and (e.g. the event)
   c. Otherwise add the message segments to he queue for the PTV. If the first segment is at the head of the queue, transmit the message segment and set the Message-try-count to zero.
6. If the message is an SNMP-encoded message of length less than $N_{MAXPACKET}$:
   a. Encode the message in an SNMP message wrapper
   b. Determine if the addition of the message to the queue for the destination PTV would cause the queue length to exceed $N_{CTLPTVQ}$, and if so discard the message and log the event
   c. Otherwise add the message to the queue for the PTV. If the first segment is at the head of the queue, transmit the message and set the message-try-count to zero

Alternative Queue Overflow Procedure:

In steps 4.b, 5.b, and 6.b above the TCIP Polling Controller may optionally handle the queue overflow situation by sending all messages in the queue (in order) followed by the new message (or all message segments) leaving the queue empty. This alternative has the advantage of giving every message an opportunity to be successfully transmitted (albeit with acknowledgements/retries) and clears the queue so that message delays are controlled.

The disadvantage of this alternative is that if many queues (to different PTVs) overflow in a short interval of time, a substantial percentage of the channel capacity can be consumed dumping overflowed queues.

10.6.14 Receiving a TCIP Message for Delivery to a Group of PTVs.

The Polling Controller upon receiving a message from an agency network for transmission to a group of PTVs shall:

1. Determine if the message length exceeds NMAXPACKET, or if the message in SNMP-encoded, and if so discard the message and log the event. Commentary: SNMP group addressing and multipacket group addressing are not supported.
2. Verify that a group has been initialized with the indicated IP address. If the group does not exist discard the message and log the event.
3. Assign the value ((last-message-sent)+1) modulo 256 to the message number field in the message wrapper, and set the last-message-sent value for the group to the same number.
4. Transmit the message in a narrowband message wrapper three successive times. Commentary: Three transmission increases the probability of success. This is implemented because group messages are not acknowledged.

10.6.15 Polling Controller Start Up

Upon startup the polling controller shall delete all allocations, and log the startup event. The polling controller shall send twenty (20) sequential Allocation Update data structures indicating that all slots are deallocated due to a restart. The polling controller shall then send a Session Poll. The polling controller shall continue to send Session Polls until a Join Request is received.

After the first Join Request is received, the polling controller shall send an Allocation Update, and resume sending Session Polls. The polling controller shall transmit only Session Polls and Allocation Updates for the first $T_{SESSIONONLY}$ seconds following the receipt of the first Join Request. At the end of $T_{SESSIONONLY}$ the polling controller shall begin the polling cycle, however for he first $T_{STARTUP}$ minutes after startup, the polling controller shall issue a Session Poll within every $T_{SESSIONPOLLSTART}$ interval (rather than the normal $T_{SESSIONPOLL}$ interval). The requirement to issue a Session Poll every polling cycle shall be in effect throughout the startup period.

Commentary: This faster Session Poll generation rates during $T_{SESSIONONLY}$ and T_{START} expedite the process of allocating slots to all PTVs in a large fleet after a polling controller restart.

10.7 VLU Rules of Procedure

10.7.1 Transmitting on the Channel

The VLU shall not transmit on the radio channel, (except to transmit a Join Request in response to a Session Poll), until it receives an Allocation Update from the polling controller indicating that the PTV has been allocated a slot number.

Once a slot number is obtained, the VLU shall transmit on the channel only:

- In response to a poll from the polling controller containing the PTVs assigned slot number
- In response to a Priority Poll while the VLU has a high-priority message queued.

10.7.2 Joining the Channel

The VLU shall join the channel as described below unless the VLU receives an Allocation Update indicating a polling controller restart. In the event of a polling controller restart the VLU must use the process defined in section

The normal process for Joining the channel is:

- The VLU shall set the message Try Count last-message-number-sent GroupID, and last-message-number-received values to zero
- The VLU shall select a random number, N_{SKIP}, in the range 0-N_{RANDOM}
- The VLU shall monitor the radio channel and skip the first N_{SKIP} Session Polls, (none if N_{SKIP} = 0)
- The VLU shall transmit a Join Request in response to the next Session Poll.
- The VLU shall monitor the radio channel for 3 seconds awaiting an Allocation Update from the polling controller.
- If no Allocation Update is received within 2 seconds, or if an Allocation Update is received that does not allocate a slot to the PTV, the VLU shall restart the join process at the first bullet above.
- If an Allocation Update is received allocating a slot to the PTV, the VLU shall begin operating on the channel using the allocated slot number.

10.7.3 Polling Controller Restart

If the VLU receives a Allocation Update indicating a polling controller restart (at any time), the VLU shall immediately discard any assigned slot allocation, and abandon any ongoing attempt to join the channel (see 10.7.2 above).

The VLU shall:

- Select a new skip count based on the formula N_{SKIP} = (vehicle ID) modulo ($N_{FLEETSIZE}$) where vehicle ID is the PTV's value of type CPT-VehicleID.
- Monitor the radio channel and skip the first N_{SKIP} Session Polls (none if N_{SKIP}=0)
- Send a Join Request in response to the next Session Poll.
- Monitor the radio for 2 seconds awaiting an Allocation Update from the polling controller.
- If an Allocation Update is received within 2 seconds allocating a slot to the PTV, the VLU shall begin operating on the channel using the allocated slot number. This would complete this process, otherwise continue with the next bullet.
- Assign N_{SKIP} = $N_{FLEETSIZE}$
- Monitor the channel and skip the first N_{SKIP} Session Polls
- Send a Join Request in response to the next Session Poll
- Monitor the radio channel for 2 seconds awaiting an Allocation Update from the polling controller
- If an Allocation Update is received within 2 seconds allocating a slot to the PTV, the VLU shall begin operating on the channel using the allocated slot number. This would complete this process, otherwise begin the Joining the Channel process defined in section 10.7.2.

10.7.4 Responding to a Normal Poll

Upon receipt of a normal poll with a slot number matching the slot number allocated to the PTV, the VLU shall:

- Determine if it should attempt to leave the channel (e.g. vehicle shutdown in progress) . If so transmit a Leave Request, and discard the slot allocation for the PTV.
- Determine if there is a message to the polling controller at the head of the queue and if so:
  - Determine if the poll acknowledges the message
  - If the poll acknowledges the message, discard the message, set message Try Count to zero, bring the next queued message (if any) to the head of the queue and assign the next send message number.
  - If the poll does not acknowledge the message and Message-try-count > = $N_{MSGMAXTRIES}$, discard the message, set Message-try-count to zero bring the next queued message (if any) to the head of the queue and assign the next send message number.
  - Otherwise leave the message at the head of the queue
- If there is a message at the head of the queue to be sent to the polling controller then ID=A3H, otherwise ID=A2H
- Generate and transmit a Poll Response using the ID from above, and containing the available fields requested in the poll. Note: a poll may ask for alarms or agency-data and there may be none to include.

- If the ID=A3H, transmit a message wrapper containing the message at the head of the queue, and increment Message-try-count.
- Update the stored GroupID value for the PTV based on the groupID received in the poll.

10.7.5 Discarding a Slot Allocation

When the VLU discards a previously provided slot allocation for the PTV, it shall discard any queued messages for the polling controller, and take any related manufacturer-defined recovery procedures and reset the Message-try-count the Last-message-number-sent, and Last-message-number-received values to zero.

10.6.7 Responding to a Priority Poll

Upon receipt of a Priority Poll, the VLU shall determine if it has any high-priority messages to send to the polling controller. If so the VLU shall respond with a Join Request containing the PTV's slot allocation, and reset the flag indicating that a high-priority message is waiting. Note: The assignment of messages to a high-priority status is a local agency decision.

10.7.7 Receipt of an Allocation Update

If the VLU receives an Allocation Update containing a polling controller restart indication, it shall conform to section 10.7.3.

If the VLU receives an Allocation Update containing an allocation of the PTV's allocated slot number to another PTV, or containing a deleted slot matching the PTV's allocated slot number, the VLU shall discard the slot allocation and begin the Joining the Channel process as defined in section 10.7.2

If the VLU receives an Allocation Update allocating the PTV to another slot (adds PTV with new slot number), the VLU shall discard the existing slot number and join the channel with the newly assigned slot number.

10.7.8 Receipt of a Message Wrapper

If the VLU receives a narrowband or SNMP message wrapper containing the PTV's allocated slot number, the VLU shall:

- Determine if there is a message to the polling controller at the head of the queue, and if so:
  - Determine if the received message wrapper acknowledges the message
  - If the wrapper acknowledges the message, discard the message, set Message-try-count to zero, bring the next queued message (if any) to the head of the queue and assign the next send message number
- Determine if the message is a duplicate (message number in wrapper matches Last-received-message-number); discard the message if a duplicate
- If the message is not a duplicate deliver it to its destination within the PTV.

If the VLU receives a packetized message wrapper containing the PTV's allocated slot number, the VLU shall:

- Determine if there is a message to the polling controller at the head of the queue, and if so:
  - Determine if the received message wrapper acknowledges the message
  - If the received wrapper acknowledges the message, set Message-try-count to zero, bring the next queued message (if any) to the head of the queue and assign the next send message number.
- Determine if the received wrapper is a duplicate (message number in wrapper matches Last-received-message-number);discard the received wrapper if a duplicate.
- Determine if the received wrapper is segment 1 of a new message. If so discard any precious multi-segment message being reassembled, and flag the message wrapper as part of the current message being reassembled.

- Determine if the received wrapper is the next segment of a multi-segment message in progress. If so add the new segment to the message being assembled. If the message is complete, deliver it to its destination in the PTV.
- Determine if the received wrapper is an out of sequence segment of a multipacket message. If so discard the current multi-segment message in progress (if any) and the received message wrapper.

If the VLU receives a narrowband message wrapper containing the PTV's stored GroupID in the slot number field (upper octet=0, lower octet matches Group ID), the VLU shall:

- Determine if the received wrapper is a duplicate (message number in wrapper matches Last-received-message-number of the group), discard the message wrapper if a duplicate
- Otherwise set Last-received-message-number for the group to the message number value from the wrapper and forward the message to its destination within the PTV.

10.7.9 Receipt of Generation or a Message to Send

Upon generating a message to send to the polling controller, or receipt of such a message from another onboard component, the VLU shall:

- Determine if the queue length equals $N_{PTVCTLQ}$, and if so discard the message and perform any necessary error recovery.
- Otherwise add the new message to the end of the queue
- If the new message is at the head of the queue, assign the next message number, and set the Message-try-count to zero

10.7.10 Receipt of Generation or a High Priority Message to Send

Upon generating a high-priority message to send to the polling controller, or receipt of such a message from another onboard component, the VLU shall:

- Determine if the queue length equals $N_{PTVCTLQ}$, and if so, discard the last message in the queue and perform any necessary error recovery
- Add the message to the front of the queue, assign the next sequential message number and set the Message-try-count to zero
- Set a flag indicating that a high-priority message is awaiting transmission to trigger the VLU to respond to the next priority poll.

Commentary: Local agencies determine which messages are to be considered high-priority for this purpose. Two examples that an agency might choose are CcOperatorCallRequest, and ImSilentAlarm.

10.8 TCIP Polling Protocol Parameters

Parameter	Description	Formula	Default
$N_{AGENCYDATAMAXTRIES}$	Number of tries to deliver agency data	Locally defined parameter	5
$N_{ALLOCRETRY}$	Number of sends on Allocation Updates	Locally defined parameter	2
$N_{BITRATE}$	Number of bits per second transmitted on the radio channel	Locally defined parameter	4800
$N_{BITSYNC}$	Number of AAH bit sync octets to send at the beginning of each transmission.	Locally defined parameter	1
N_{SKIP}	The number of Session Polls to be skipped by a VLU before generating a Join Request. This parameter reduces the rate of Join Request Collisions between PTVs.		
$N_{PTVCTLQ}$	Length of message queue to the polling controller in each allocated PTV	Locally defined parameter	8
$N_{CTLPTVQ}$	Length of message queue to each allocated PTV	Locally defined parameter	8
$N_{MAXBADPOLL}$	Number of bad polls before deallocating a slot	Locally defined parameter	10
$N_{MAXMSGLENFROMPTV}$	Maximum number of octets in a narrowband encoded TCIP message from a PTV	Locally defined parameter	100
$N_{MAXMSGLENTOPTV}$	Maximum number of octets in a narrowband encoded TCIP message to a PTV	Locally defined parameter, Cannot Exceed 15* $N_{MAXPACKET}$	500
$N_{MAXPACKET}$	Maximum number of octets in a segment of a multi-segment controller to PTV message.		300
$N_{MSGMAXTRIES}$	Number of tries to deliver a message wrapper	Locally defined parameter	5
$N_{FLEETSIZE}$	Approximate peak number of PTVs sharing the radio channel.	Locally defined parameter.	1200 -- illustrative only
N_{RANDOM}	Highest value in random number range used to randomize access to Session Polls.	Locally defined parameter	30
$T_{MESSAGEWAIT}$	Wait in milliseconds for a message to be sent following a poll response (ID=A3H).	$T_{RADIOTIME} + [(\{12+N_{BITSYNC}+ N_{MAXMSGLENGTH}\}*8) *(1000/N_{BITRATE})]$	200
T_{PRMAX}	Time in milliseconds to wait for a poll response (possibly containing agency data or alarms) before declaring a dry poll.	$T_{PRMED}+[(116*8)*(1000/N_{BITRATE})]$	317
T_{PRMED}	Time in milliseconds to wait for a medium poll response (optional fields, but no alarms or agency data) before declaring a dry poll.	$T_{PRMIN}+[(36*8)*(1000/N_{BITRATE})]$	122
T_{PRMIN}	Time in milliseconds to wait for a minimal poll	$T_{RADIOTIME} + [(\{13+N_{BITSYNC}+ 17\}*8) *(1000/N_{BITRATE})]$	62

	response (no optional fields) before declaring a dry poll.		
$T_{RADIOTIME}$	Time in milliseconds added to each poll wait to allow for radio and VLU latency	Locally defined parameter	10
$T_{FASTPOLLINTERVAL}$	Maximum time in seconds between normal polls to a PTV in fast poll mode	Locally defined parameter	20
$T_{PRIORITYPOLL}$	Maximum time in seconds between priority polls	Locally defined parameter	5
$T_{SESSIONONLY}$	Length of interval in seconds after receipt of the first Join Request during which only session polls and no priority or normal polls are issued.	Locally Defined parameter	60
$T_{SESSIONPOLLSTART}$	Maximum time in seconds between Session Polls during startup.		2
$T_{STARTUP}$	Time in minutes, after a polling controller restart during which Session Polls are sent at a higher than normal rate	Locally defined parameter. Suggest a minimum value of (PTV count)*$T_{SESSIONPOLLSTART}$*2/60	40
$T_{SESSIONPOLL}$	Maximum time in seconds between Session Polls.	Locally defined parameter	8
$T_{SESSIONWAIT}$	Wait in Milliseconds after a Session Poll before declaring a dry poll.	$T_{RADIOTIME}$ + ({8+$N_{BITSYNC}$}*8)*(1000/$N_{BITRATE}$)	35