MOTOTRBO™

Data Services Overview

Version 01.14

July 9, 2012

COPYRIGHTS

REVISION HISTORY

Version	Date	Section	Page	Lines	Description
01.09	08/31/2011	1.4 4.2	5 27	20 418-427	**CCMPD01307529** Added new acronyms
		1.5	6	38	**CCMPD01307529** Added BT ADK Guide to the references section.
01.10	08/31/2011	1.5	6	38	**CCMPD01492794** Removed reference to deprecated document (USB configuration Guide)
01.11	09/23/2011	5.0	32	668-671	**CCMPD01530668** Enhance the Data revert rule in MOTOTRBO R1.8A
01.12	12/20/2011	3.4	14	194-206	**CCMPD01573230** Add MOTOTRBO R2.x/1.x differnce on IP address and ICMP
01.13	05/04/2012	1.4 3.2 4.1 4.2	5, 6, 10 – 12, 24 - 26, 28 - 32	21, 115 – 147, 449 – 524, 540 – 596, 608, 640, 649 – 674, 695 - 721	**CCMPD01648715** Private Area Network information using the Expanded Portfolio & SL Series subscribers as Bluetooth Personal Area Network access points.
01.14	06/18/2012	3.2	10	122 - 129	**CCMPD01667213** Added notations on IPv4 Network ID address limitations for core subscribers and for enhanced portfolio subscribers running early R2.0 FW on over-the-air data transmissions with Bluetooth PAN devices.
		3.5	17	236 – 238	**CCMPD01515150** Added notation on required value in IP header's TTL field for Enhanced Portfolio and SL Series radios.
		All	All	All	Changed terminology: "Expanded Portfolio" is now "Enhanced Portfolio". "Legacy" MOTOTRBO radios are now "Core" MOTOTRBO radios.

TABLE OF CONTENTS

Section 1

1.0 Introduction

1.1 Overview

MOTOTRBO™ is Motorola Solutions' new Professional Radio that operates in both analog and digital modes. In digital mode, in addition to digital enhancement of the voice, MOTOTRBO™ also provides a pipe for transmitting data packets between data applications.

An application has often two parts, a client and a server. An application's client can reside in a radio, or in an Option board, or in a laptop connected to the radio. MOTOTRBO™ has four built-in application clients: Location, Text Messaging, Telemetry, and ARS client. They are integral part of the radio software. MOTOTRBO™ does not support downloading of an application into the radio's memory.

1.2 Scope and Purpose

The main emphasis of the document is to describe, at a high level, the interfaces and the services supported by the data pipe. The information is intended for use by Application Developers. Note that the document does not describe the application layer interfaces supported by the built-in applications.

1.3 Assumptions

The document assumes that MOTOTRBO™ radio is in digital mode.

1.4 Terminology

Acronyms	Description
ACK	Acknowledgement
ADK	Application Development Kit
ARS	Automatic registration service (also called as Presence Service)
AT	Access Type
BT	Bluetooth
CACH	Common Announcement Channel
CAI	Common Air Interface
Core MOTOTRBO Subscriber	Original model MOTOTRBO mobile & portable radios
CPS	Customer Programming Software
CRC	Cyclic Redundancy Checksum
CSBK	Control Signaling Block
DHCP	Dynamic Host Configuration Protocol
DMR	Digital Mobile Radio, a standard
Enhanced Portfolio MOTOTRBO Subscriber	MOTOTRBO mobile & portable radio models with optional color displays and Bluetooth capabilities.
ETSI	European Telecommunications Standards Institute
IP	Internet Protocol

Acronyms	Description
ICMP	Internet Control Message Protocol
NACK	Negative Acknowledgement
NAT	Network Address Translation
PAN	Personal Area Network (Bluetooth Profile).
PAN-NAP	Personal Area Network – Network Access Point (Bluetooth Profile). The MOTOTRBO Enhanced Portfolio & SL Series subscribers serve as PAN-NAP devices.
PANU	Personal Area Network User (Bluetooth Profile). Only devices defined as PANU devices may connect to the Enhanced Portfolio & SL Series radios via Bluetooth PAN.
PTT	Push-To-Talk
SACK	Selective Acknowledgement
SPP	Serial Port Profile
TDMA	Time Division Multiple Access
TCP	Transmission Control Protocol
UDP	User Datagram Protocol
XCMP	Extended Control And Management Protocol

1.5 References

[1] ETSI TS 102 361-1 Electromagnetic compatibility and Radio spectrum Matters (ERM); Digital Mobile Radio (DMR) Systems; Part 1: DMR Air Interface (AI) protocol

[2] ETSI TS 102 361-3 Electromagnetic compatibility and Radio spectrum Matters (ERM); Digital Mobile Radio (DMR) Systems; Part 3: DMR Data protocol

[3] MOTOTRBO™ Option Board ADK Guide

[4] MOTOTRBO™ XCMP/XNL Development Guide

[5] MOTOTRBO™ XCMP/XNL Development Specification

[6] MOTOTRBO™ Telemetry ADK Guide

[7] MOTOTRBO™ Telemetry Protocol Specification

[8] MOTOTRBO™ Location Data ADK Guide

[9] MOTOTRBO™ Location Request and Response Protocol Specification

[10] MOTOTRBO™ Text Messaging ADK Guide

[11] MOTOTRBO™ Text Messaging Protocol Specification

[12] Presence Notifier Application User’s Guide

[13] Presence Notifier-to-Watcher Interface Specification

[14] Multi-Channel Device Driver (MCDD) Configuration Guide

[15] MOTOTRBO™ ARS Protocol Specification

[16] MOTOTRBO™ Non-IP Capable Peripheral ADK Guide

[17] MOTOTRBO™ System Planner

[18] MOTOTRBO™ Bluetooth ADK Development Guide

Section 2

2.0 Overview of the Data Services

Applications are the users of the Data Services. Applications are of two types: internal and external. The internal applications reside inside a radio and external applications are outside on a peripheral connected to the radio. A peripheral can be either an IP capable (e.g. a laptop) or a non-IP capable (e.g. an option board).

The data services provide an application a way to transmit and receive data messages to/ from another application, where the other application can be either attached to the same radio or to a different radio. In case the other application is attached to another radio, the data services use the radio network to send/receive data messages.

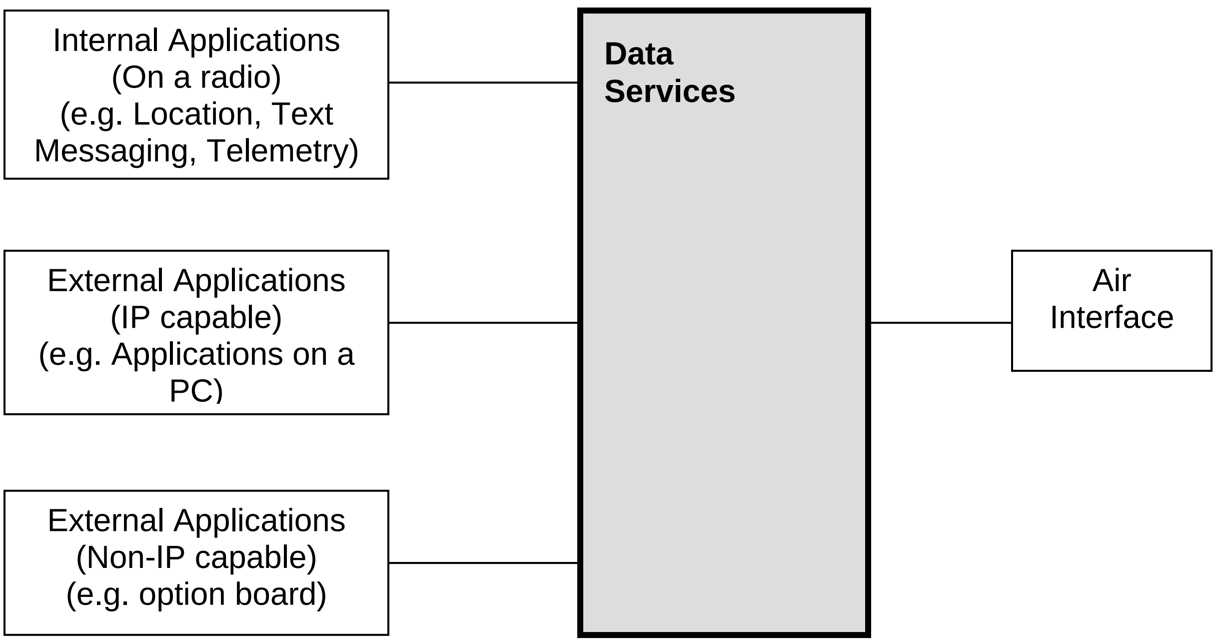

Figure 1 – Context Diagram of Data Services

Section 3

3.0 Interface to an Internal Application

Applications residing on a radio use the IPv4 protocol stack of the radio to communicate with the other applications. Each radio has a unique IPv4 address and the multiple applications on the radio are distinguished by the applications' unique UDP or TCP port number. All the applications on a radio have the same IPv4 address, which is the IPv4 address of the radio.

3.1 Transport Layer: UDP /TCP interface

MOTOTRBO™ provides a unique UDP or TCP port number to each internal application. The port numbers are used to route an incoming message to its destination application. Use of UDP is preferable when messages are communicated over the air. The port numbers of internal applications are constants and are not programmable using CPS. This is to avoid assigning port numbers incorrectly by the User.

A MOTOTRBO™ radio also has a proxy application that facilitates data communication to or from a non-IP capable device (i.e. an Option Board connected to the radio). There are two types of non-IP capable device, which are Option Board and Non-IP Capable Peripheral. The proxy application provides four UDP ports for these two types of non-IP capable device respectively. For more details, refer to section 5.0.

The table below provides the port numbers of the internal applications:

Application Name	UDP port number	TCP port number
Location Client	4001	-
Text Messaging Client	4007	-
Telemetry	4008	-
ARS or Presence Service Client	4005	-
Option board Text Message	4061	-
Option board Telemetry	4062	-
Option board Location Data	4063	-
Option board Raw Data	4004	-
Non-IP Capable Peripheral Text Message	4066	-
Non-IP Capable Peripheral Telemetry	4067	-
Non-IP Capable Peripheral Location Data	4068	-
Non-IP Capable Peripheral Raw Data	4069	-
Bluetooth SPP Raw Data	4072	
DHCP Server	67	-
DHCP Client	68	-
XCMP		8002

NOTE:

1. The radio checks the Checksum field of the UDP and TCP headers.. In case of failed checksum, the radio discards the packet and does not send an ICMP message because the destination address might be corrupted.
2. After a TCP connection is closed, it is possible that some datagrams are still making their way. TCP has a “quiet timer” that prevents the just closed port from reopening and receiving those datagrams.

3.2 Network Layer: IPv4 Address of a Radio & its IPv4 Compatible Peripheral

The IPv4 address of a radio has a simple one-to-one mapping with the radio’s Radio ID. The derivation of IP addresses simplifies the configuration of a radio. It also eliminates the need for implementation of the Address Resolution Protocol (ARP). The Radio ID is also used to derive an IPv4 address of an IP capable peripheral (e.g. a laptop) connected to the radio.

The derivation of the IPv4 addresses of the radio and its IPv4 capable peripheral follows the following rules: (Refer to Figure 2.)

- The IPv4 address of radios and their IPv4 capable peripherals are Class A addresses.
- The “Host Number” field of the IPv4 address of a radio or its peripheral is same as the Radio ID. The Radio ID can be configured through CPS as shown in Figure 3.
- The "Network Id" field of the IPv4 address of a radio can be configured using CPS, which is the CAI Network field shown in Figure 4 (default value = $0C).
- The “Network Id” field of the IPv4 address of an IPv4-capable USB peripheral of the radio is the “Network ID” field of the radio + 1.
- The “Network Id” field of the IPv4 address of an IPv4 capable Bluetooth device connected to the radio is the “Network Id” field of the radio + 2.

IMPORTANT: Core MOTOTRBO subscribers are designed to send data over the air to, or receive data over the air from, remote devices with IPv4 “Network Id” values equal to:

- The CAI Network field value (default value = $0C)
- The CAI Network field value + 1

Neither core MOTOTRBO subscribers nor enhanced portfolio MOTOTRBO subscribers running firmware older than R02.04.00 permit the transmission of data to a “Network ID” value equal to (CAI Network field value + 2). Nor will the aforementioned subscribers receive data from a “Network ID” value equal to (CAI Network field value + 2).

The above rules are shown graphically in the figure below:

Class A	0	XXXXXXX	Radio ID
	Network Id (8 bits)		Host Number (24 bits)

Figure 2 – Derivation of an individual IPv4 address of a radio.

For example, in **Figure 3** the Radio ID is set to 1234. The Network Id is set to 12 as shown in **Figure 4**. The steps below show how to calculate the radio's IP address:

1) Convert the Radio ID to Hex: 0x0004D2.
2) Separate into 8-bit section: 0x00 0x04 0xD2.
3) Convert each section into decimal: 00, 04, 210.
4) Each 8 bit section represents 1 byte of the IP address, so the radio's IP address is 12.00.04.210.

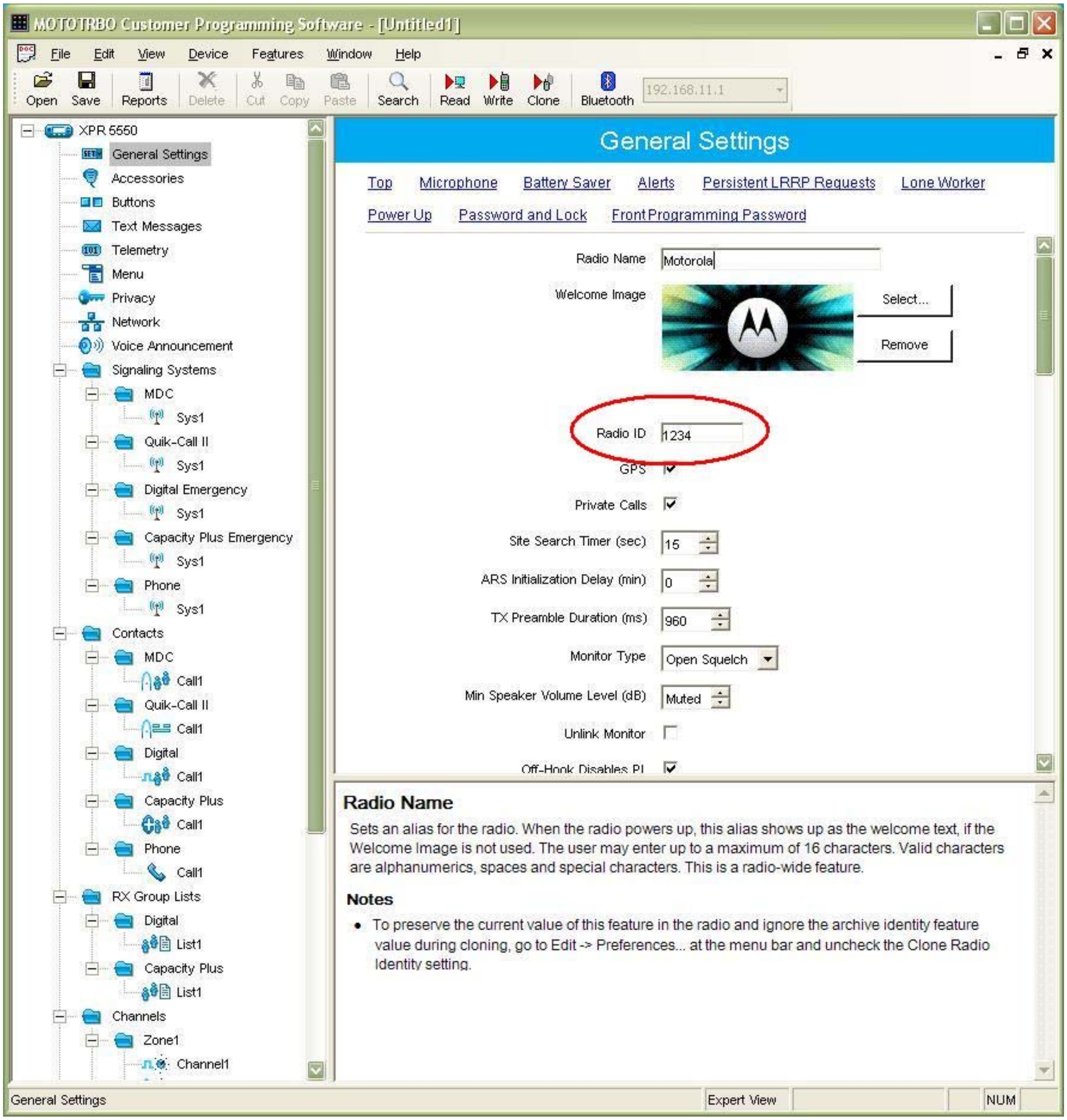

Figure 3 - MOTOTRBO™ Radio's Radio ID Configuration Screen on CPS

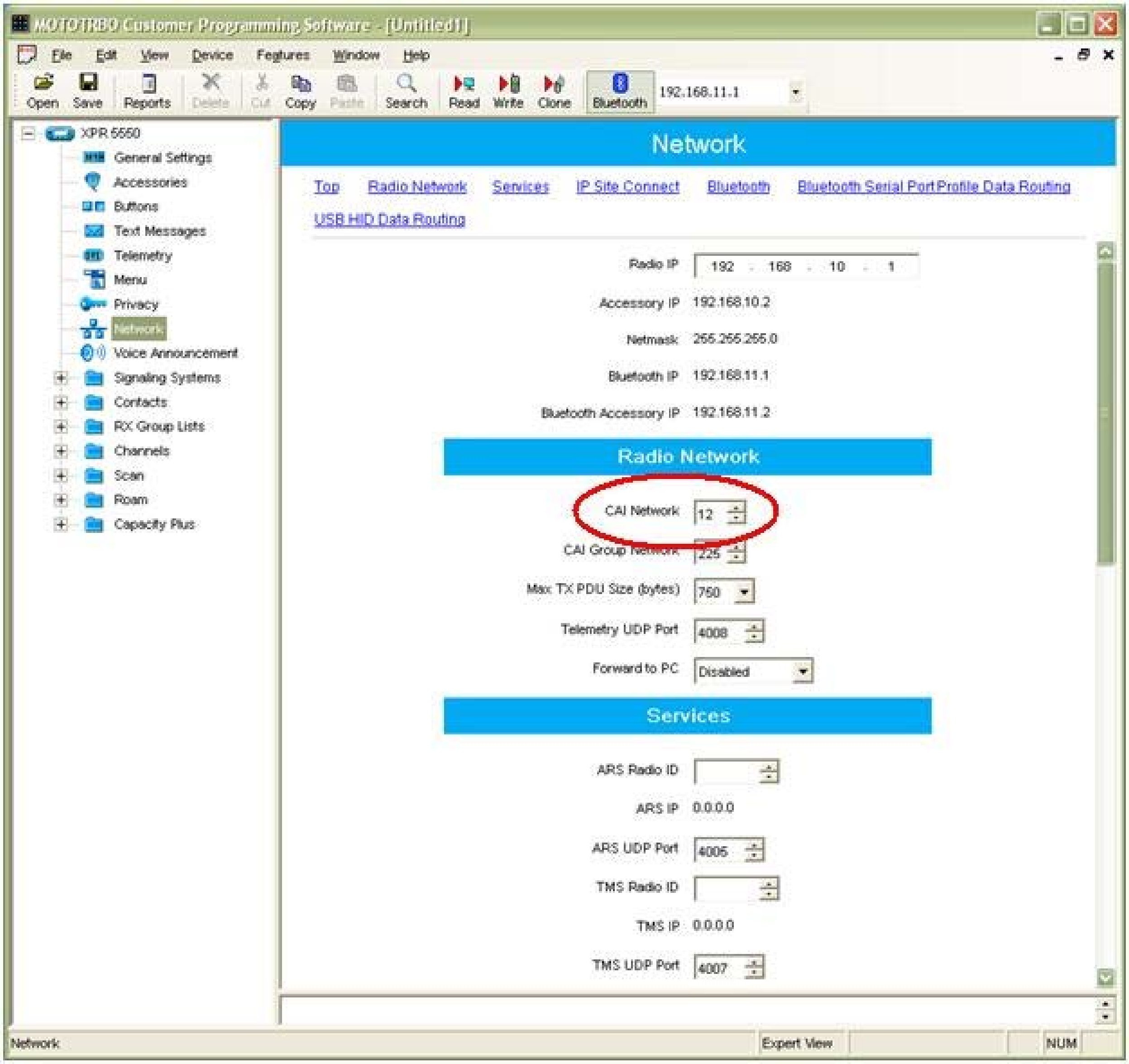

Figure 4 – MOTOTRBO™ Radio's Network Configuration Screen on CPS

3.3 Network Layer: IPv4 Address of a Group of Radios

A unique service offered by the MOTOTRBO™ data services is sending an IPv4 data message to a group of radios, where the group is a “Talkgroup”. The IPv4 address of a group is derived from the layer 2 address of the talkgroup using the following rules:

- The most significant 8 bits of the IPv4 address of a group is class D address with the most significant 4 bits set to E_{16}.
- The least significant 24 bits of the IPv4 address of a group is same as the layer 2 address of the talkgroup.
- The "Network Id" field of the IPv4 address of a radio can be configured using CPS, which is the CAI Group Network field shown in 4. (default value = $E1).

Note

Capacity Plus supports 8 bit talkgroups 1-254; 255 is reserved.

The above rules are shown graphically in the figure below:

Class D	1110	XXXX	24 bit of Layer 2 address of a group
	Network Id (8 bits)		

Figure 5 – Derivation of a group IPv4 address of a radio.

NOTE: The group messaging service of MOTOTRBO™ is not based on any IP multicasting protocol. It is based on the broadcast nature of the air interface and a radio’s capability to receive a transmission directed to a talkgroup whose member the radio is. A radio does not duplicate a received group message and therefore it is received either by an application in the radio or by an application in the peripheral attached to the radio.

3.4 Network Layer: IP Error Messages

The Internet Protocol (IP) does not provide end-to-end data reliability but it does report an error in datagram processing. The Internet Protocol uses the Internet Control Message Protocol (ICMP) for reporting errors.

Typically ICMP messages are 36 octets long. ICMP messages are sent using the basic IPv4 header. It always includes the IP header (and options) along with the first 8 bytes of data from the initial fragment of the IP datagram that caused the error. The purpose of ICMP messages is to provide feedback about problems in the communication environment, not to make IPv4 reliable. There is still no guarantee that a datagram will be delivered or an ICMP message will be returned. Some datagrams may still be undelivered without any report of their loss. To avoid the infinite regress of messages about messages etc., no ICMP messages are sent about loss of ICMP messages.

There are two classes of ICMP messages: errors and queries. Query messages are defined in pairs: a request and a reply. ICMP query messages are useful for diagnosis and troubleshooting. Note that the “Ping” program uses ICMP query messages. Both MOTOTRBO™ radio and repeater return an ICMP echo response upon reception of an ICMP echo request message.

The following table lists the ICMP error messages generated by a radio.

ICMP Message Name (Type)	Code	Comments
Destination Unreachable (Type = 3)	Network is Unreachable (code = 0)	The IP datagram is destined for the attached PC’s IP address and 1. the PC is not present or 2. the PC does not support IP or 3. the PC is in the process of removal from the radio.
	Host is Unreachable (code = 1)	1. The sending radio exhausts the maximum number air interface retry attempts, or 2. The maximum queue size for datagrams would have been exceeded by the received datagram, or 3. The maximum queue dwell time for the datagram has been exceeded.
	Fragmentation is needed but DF is set. (code = 4)	This is generated when an IP datagram is received from either the RF network or from a connected PC with the "Do Not Fragment" bit set but the size of the datagram exceeds the maximum transfer unit (MTU).
	Destination network is unknown. (code = 6)	A properly formatted IP datagram is received by a radio indicating a destination network class that is not supported by the radio.
Source Quench (Type = 4)	No buffer space is available to queue the datagram. (code =0)	The source quench message is a request issued to decrease the traffic rate of the data messages to the internet destination. The traffic reduction prevents buffer space overflow. (When the buffer space overflows or becomes congested, the datagrams are lost or discarded.)
Parameter Problem (Type = 12)	IPv4 Header is bad. (code = 0)	This is generated by a radio when it receives an IP datagram with improper formatting of its IP header and does not conform to IP version 4 format.

The table below lists the ICMP error messages generated by a Repeater.

ICMP Message Name (Type)	Code	Reject Reason
Destination Unreachable (Type = 3)	Host is Unreachable (code = 1)	The maximum queue size for datagrams would have been exceeded by the received datagram,
	Fragmentation is needed but DF is set. (code = 4)	A properly formatted IP datagram is received by the repeater which exceeds the maximum transfer unit (MTU) and the datagram has the "Do Not Fragment" bit set in the IP header.
Parameter Problem (Type = 12)	IPv4 Header is bad. (code = 0)	This is generated by a repeater when it receives a datagram with a bad IPv4 header.

NOTE: The ICMP packet’s payload length in MOTOTRBO R1.x is fixed because there it includes the header of the oringnally sent message in ICMP payload. The ICMP payload length varies for Enhanced Portfolio and SL Series MOTOTRBO R2.x radios, because the originally sent message is included in its entirety.

The ICMP packet’s TOS(Refer to section 3.5) bits in MOTOTRBO R1.x is 0x00 but in MOTOTRBO R2.x it is 0xC0.

3.5 Network Layer: Priority Data, Immediate Data and Unbuffered Feature

The Data Application can request priority treatment of data messages and Immediate Data independently. It also can control the data if it is send out with or without the queue. To facilitate those, the Type Of Service (TOS) bits in the Internet Protocol (IP) header will be used to request this special processing of the data messages. The position of the TOS field in the IP header is shown in Figure 6 and is most completely discussed in RFC 791.

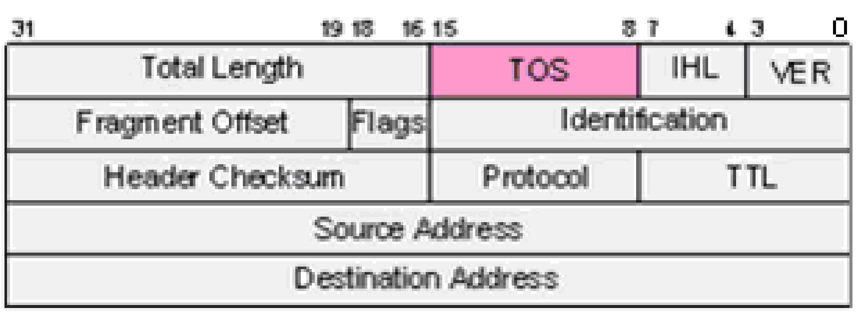

Figure 6 –Internet Header Format

The TOS field is further partitioned and contains a Precedence field and other bits as shown in Figure 7. The bit8, bit9 and bit 10(Precedence) will be used to specify whether

Normal, Priority, or Immediate processing of the IP datagram should occur. The bit 11(D), bit 12(T), bit 13(R) and bit 14 are not used by MOTOTRBO and should set to 0. Bit 15 is used to indicate Unbuffered feature for Immediate, Priority and Normal data.

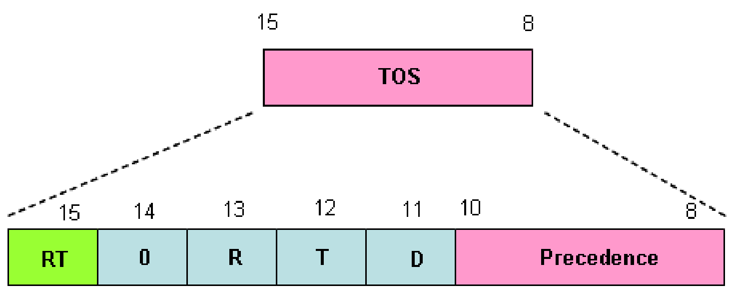

Figure 7 –Service Mappings

The mapping between the field values and the MOTOTRBO data services is shown in Table 1.

TOS	Data Type
0x00	Normal Data
0x01	Priority Data
0x02	Immediate Data
0x80	Unbuffered Data
0x82	Unbuffered Immediate Data
Others	Reserved

Table 1: Map of TOS and Data Type

Normal Data is used when a device needs to send one Data Unit across to another device.

Immediate Data is used when a device needs to send one Data Unit to another device. The message is sent with immediate interruption of a voice call or normal data call in progress

Priority Data is used when a device needs to send one Data Unit to another device. The message is sent with priority over normal data in queue. This function does not interrupt any voice call or normal data call in progress.

Unbuffered Data is used when a device needs to send one Data Unit to another device. If the data sending is blocked by channel busy for other voice call or data call, the data will not be queued but discarded immediately.

Unbuffered Immediate Data is used when a device needs to send one Data Unit to another device. The message is sent with immediate interruption of a voice call in progress. If the in-progress voice call is not interruptible, the data will not be queued but discarded immediately.

When Unbuffered Data is discarded immediately, the radio will send back the ICMP message with the Type set to Destination Unreachable (3) and Code set to Host is Unreachable (1).

Please refer to [4] for more detailed information about Priority Data, Immediate Data and Unbuffered Feature.

NOTE: IP-based applications must set the TTL (Time To Live) value to a value greater than 1 (at least 2) when connecting with MOTOTRBO Enhanced Portfolio and SL Series subscribers. Otherwise IP packets will not be sent over the air to another radio. In accordance with standard IP rules, TTL is decremented as the packet passes through a router. This includes the Enhanced Portfolio subscriber.

3.6 Data Link Layer

The data link layer of MOTOTRBO™ is compliant with the ETSI DMR Packet Data Protocol. The data link layer receives the IPv4 datagram from the network layer for transmission over-the-air as DMR Packet Data Units (PDU). The implementation of the IPv4 stack limits the maximum size of the IPv4 datagram to 1500 bytes. Considering that UDP/IP headers take 28 bytes, the maximum size of an application's message is 1472 bytes. The IPv4 datagrams from the network layer are buffered by the data link layer. The buffer has a capacity of holding 10 messages. An ICMP Destination Unreachable datagram is returned in the following error conditions:

- A datagram is received with the buffer full, or
- A buffered datagram is discarded because the datagram was in the buffer for more than 210 seconds, or
- The datagram failed to transmit over-the-air and was discarded.

The data link layer provide following important services:

- Fragmentation of a datagram into DMR Packet Data Units (PDUs) and reassembly of PDUs into a datagram,
- Confirmation of PDUs, and
- Rejection of duplicate PDUs for a datagram fragmented into multiple PDUs.

3.7 Data Link Layer: Fragmentation and Reassembly

DMR transfers a datagram over-the-air in packets. A datagram is first split into fragments, where each fragment is the information contained in a packet. The fragments are then formed into packets, and the packets are split into a sequence of information blocks. Each block is protected by error correcting code, and then the sequence of blocks is transmitted over-the-air. The maximum size of a fragment is limited by a CPS programmable parameter and it can be 300 or 500 or 750 or 1500 bytes with a default value of 500 bytes. The maximum number of fragments is 5 when the fragment size is 300 bytes.

NOTE

Capacity Plus does not support layer 2 fragmentation. The maximum datagram size, including the IP and UDP headers, are identical to the max TX PDU size specified by the CPS programmable parameter.

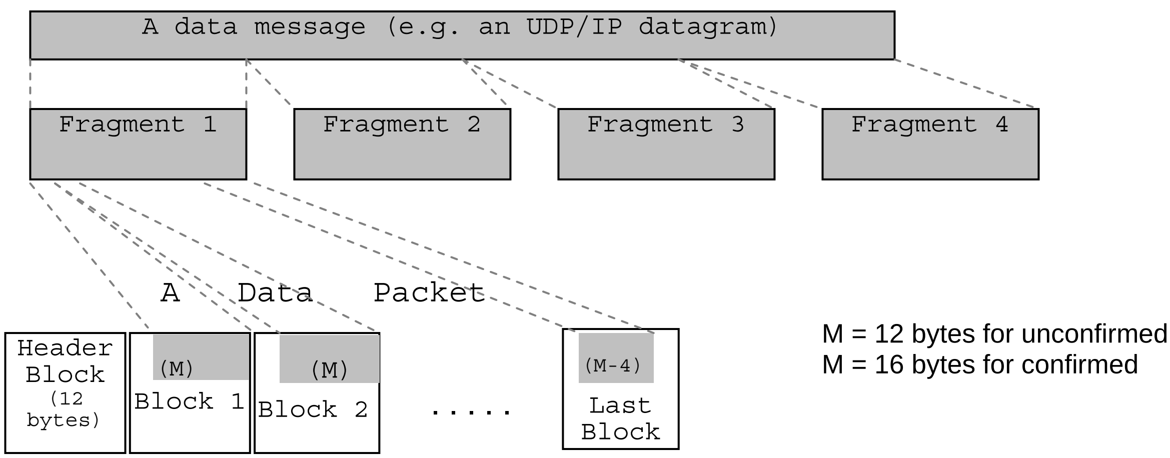

Figure 8 – Decomposition of a datagram into packets

Figure 8 shows the structure of a data packet. The details of the data packet structure are available in Reference [1] and Reference [2]. Each data packet begins with a header block followed by a set of data blocks.

To assist in assembly of fragments, the following fields are provided in the header block:

1. Fragment Sequence Number Field (FSNF) (4 bits): The three least significant bits are used for sequence number. The sequence numbers for the fragments are {0, 1..7, 1..7, ..}. The fourth bit of the FSNF is set to 1 for the last fragment.
2. Sequence number of the packet (N(S)) (4 bits): A successful transmission of a packet may require multiple transmission of the packet. This field is used to distinguish the duplicate packets.
3. Syn (1 bit): When asserted, this field synchronizes the packet and fragment sequence number between the sending and receiving Subscriber Units (SUs).

3.8 Data Link Layer: Confirmed Data

The format of the data block is different for confirmed and unconfirmed data. A confirmed data packet is more efficient than unconfirmed because each confirmed data block carries 4 more bytes of user data compared to unconfirmed data block. It is also more reliable due to acknowledgement.

MOTOTRBO is configured to transmit a private message (unicast) IP datagrams as confirmed or unconfirmed through the CPS for Release 1.5 and beyond. MOTOTRBO uses unconfirmed data packet for group messages (multicast)) IP datagram. For more information, refer to Network/Radio Network/CAI Network in the CPS.

The first two bytes of a confirmed data block has a 7-bit serial number and a 9-bit CRC. On first try, all the data blocks are sent. On receipt of the last block of each confirmed data packet, the receiver sends back a response (ACK or NACK or SACK (selective acknowledgement)).[1] If there is a CRC mismatch for some of the blocks then the receiver sends a response message containing the list of blocks whose CRC is not matching. The sender retransmits the listed blocks. This process repeats (up to a maximum of 3 times) until all the blocks are transmitted successfully.

[1] MOTOTRBO sets the A/N bit in header block of all confirmed data packets.

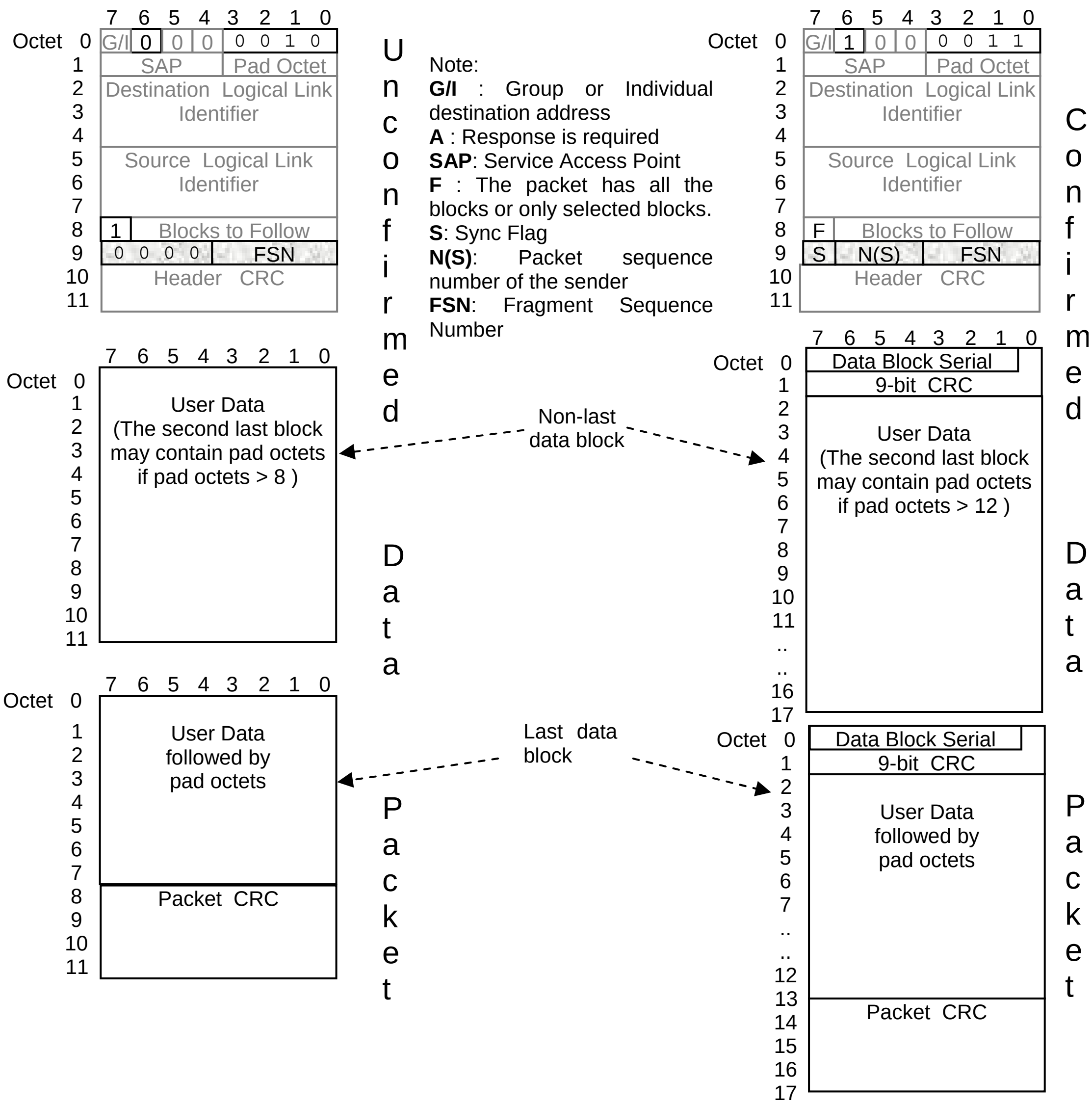

Figure 9 – Format of unconfirmed and confirmed packets

A receiving radio sends the acknowledgement (i.e. ACK or NACK or SACK) within one ‘burst’ after receiving the “last block” or terminator. In direct mode, a sending MOTOTRBO™ radio always appends one “data terminator” at the end of a sent data packet. In repeater mode, a MOTOTRBO™ repeater reserves two bursts following the “last block” for the acknowledgement by setting the slots to busy2. The repeater also indicates to the receiving radio the end of a data packet by transmitting a proprietary “Terminator with LC”. This is shown in Figure 10.

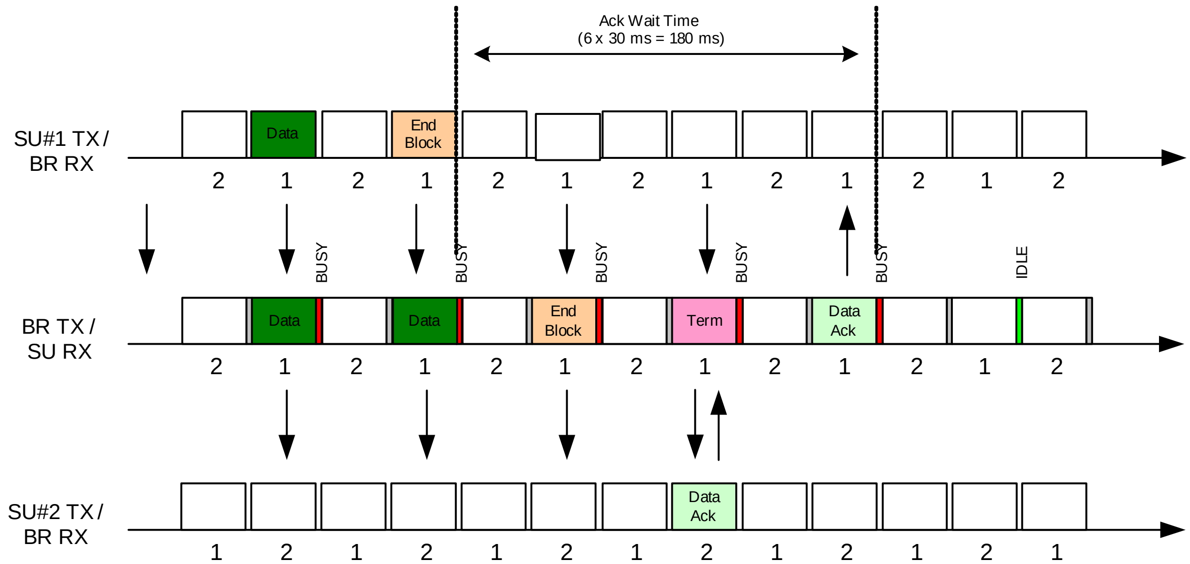

Figure 10 – Single Block ACK in Repeater mode

3.9 Data Link Layer: Multiple Headers

Normally a data packet begins with a header block. If “scan feature” is enabled, a set of preamble CSBK are added in front of the header block of a "unconfirmed data packet" (used only for group messages). The preamble improves the reliability of scanning unconfirmed data packets. The length of the preamble is configurable by CPS. The default value is 2 CSBKs.

Preambles are also added for an effective Battery Power Saving. In Talk Around mode, Battery Power Saving requires transmitting two CSBK preambles and in repeater mode it requires to transmit one CSBK preamble.

When both Battery Power Saving and Scan are active, the number of preamble CSBK is the larger of the two preambles.

[2] This is achieved by setting the ‘AT’ bit of CACH to BUSY () refer to 1.

3.10 Data Link Layer: Enhanced Channel Access

Enhanced Channel Access (ECA) feature is a channel access procedure in which the call initiating radio transmits channel access request and listens on the channel to determine the status of the request. Radio transmits the call only when the access to the channel is obtained. Only one of the requesting radios obtains the access to the channel and proceeds with call transmission while the other radios back-off.

Enhanced Channel Access provides the ability to reserve Over The Air (OTA) channel for one of the call initiating radios, and pexclusive access to that radio for a small duration.

Enhanced Channel Access feature is applicable only in repeater mode of operation and can be configurable on the radio. It can be enabled or disabled on conventional digital channel, IPSC local and wide area channels and GPS/data revert channel.

See MOTOTRBO System Planner for detailed information on Enhanced Channel Access interacts with admit criteria setting.

Enhanced Channel Access Advantages

- Improve data call success rate by minimizing Over the Air (OTA) call collisions due to multiple radios keying up very close to each other.
- Only one radio among multiple call initiating radios gets the access to the channel and proceeds with the call. Other radios do not transmit the call and can receive the ongoing call if the call is of interest.
- Prevents call transmission when the radio is out of inbound range (but within the outbound range) and provides correct call status indication to the user.
- Improves the GPS data success rate on the GPS revert channel by minimizing the collisions.

Enhanced Channel Access Limitations

Enhanced Channel Access slightly increases the latency for the data and CSBK calls.

Section 4

4.0 Interface to an IP capable External Application

Both portable and mobile radios have an interface for a device having an IPv4 stack. This interface is used to connect a PC to a radio. The following cables must be used to attach the MOTOTRBO™ radio to the PC's USB port:

- Portable – PMKN4012
- Mobile – HKN6184

Also the following software is required for proper connection of the MOTOTRBO™ radio as a USB device within Windows 2000 or Window XP PC environment:

- MOTOTRBO™ USB Device Driver

This section describes the flow of an IP datagram from/to an application in a PC attached to a MOTOTRBO™ radio.

An IP datagram received by a radio that is addressed to a talkgroup is delivered to either an internal application inside the radio or to an application on the attached PC (i.e. an external application). The received IP datagram is not duplicated and then sent to both an internal and external application.

An IP datagram received by a radio that is addressed to an internal application may need to be re-directed and routed to an external application in certain configuration (e.g., internal text messaging application versus PC-based text messaging that are mutually exclusive applications and the sender is not aware about which application is active). This routing is CPS provisionable in the radio. An IP datagram received by a radio that is addressed to an external application is always delivered to the attached PC. i.e., the IP datagram is not re-directed and routed to an internal application. A MOTOTRBO™ radio allows an application on the attached PC (e.g., CPS) to communicate with an internal application.

MOTOTRBO™ radio does not support broadcast group or system-wide group. Since the MOTOTRBO™ radio supports only a single "radio system", it is required that both the radios identity and "talkgroup" membership is consistently defined across all personalities (i.e., TGID=1 has the same membership, no matter which personality or frequency it is used on).

4.1 An Example of IP Addresses in a Radio

This section describes an IP addressing scheme. A radio supports four separate IP spaces on the Common Air Interface (CAI), one for individual addressing of internal applications, one for individual addressing of external applications via USB connection, one for individual addressing of external applications via Bluetooth PAN connection, , and one for group addressing (either internal or external). A radio is really multi-homed on four independent overlaying network segments. Additionally, since the radio may belong to multiple groups, it may have multiple presences (i.e., multiple IP addresses) on the network segment supporting group addressing: one per group of interest. In this way, group addressing is supported without needing to implement any of the Internet Group Management Protocol (IGMP) between the attached PC and the radio or over the CAI. The air interface implicitly serves as a multicast router with multicast group membership statically provisioned into every radio via CPS.

12.x.x.x represents the CAI Subscriber IPv4 Address with Network ID of 12 and Host Number derived from the Radio ID.

13.x.x.x represents the CAI IPv4 Address for the attached PC through USB connection. The Network ID is 13, and the Host Number is derived from the attached radio's ID.

14.x.x.x represents the CAI IPv4 Address for the attached PC through Bluetooth PAN connection. The Network ID is 14 and the Host Number is derived from the attached radio's ID.

225.x.x.x is the CAI Takgroup IPv4 Address with Network ID of 225 and Host Number derived from the Talkgroup ID.

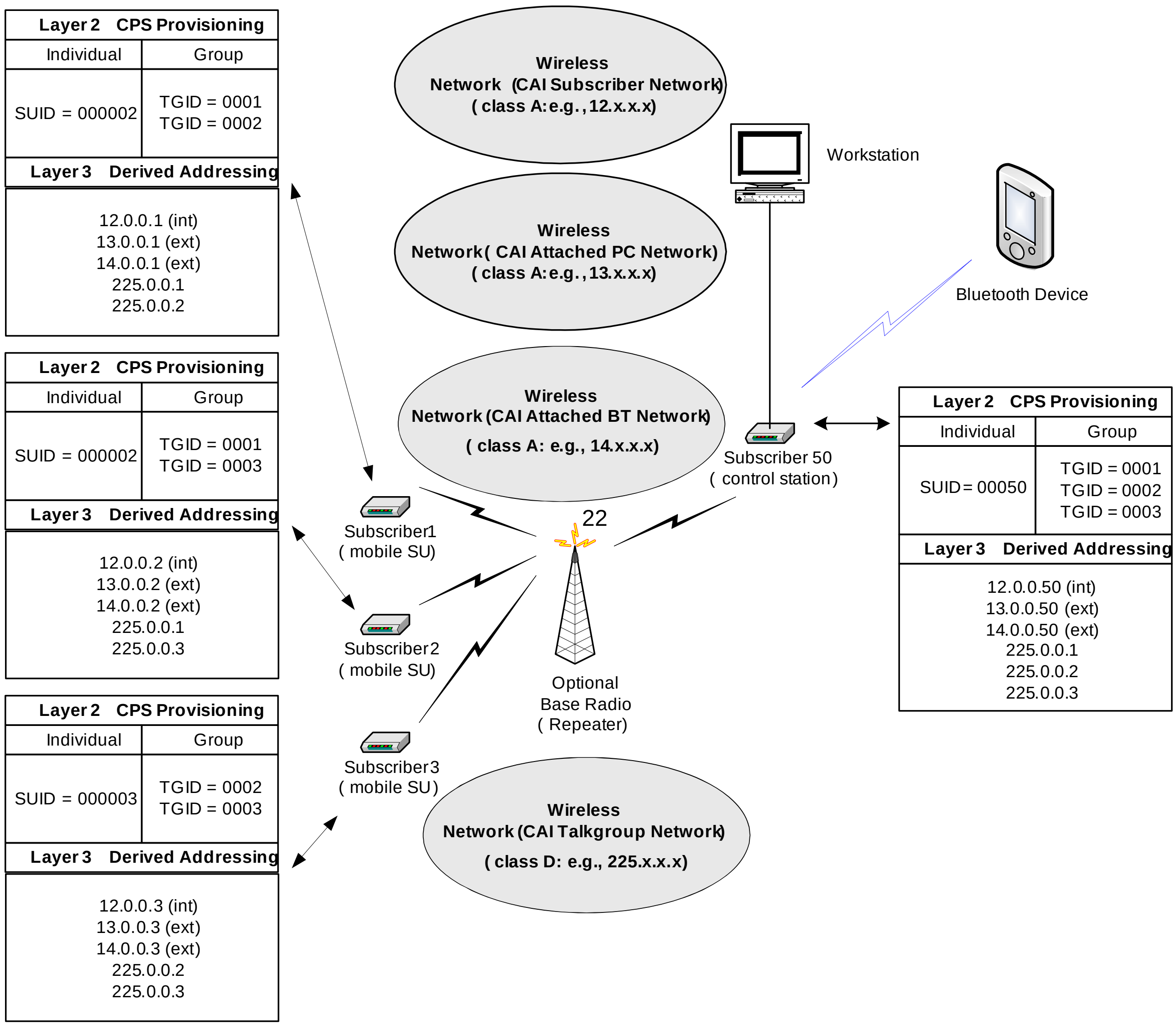

Figure 11 – An example IP address schema

Figure 11 illustrates a very simple MOTOTRBO™ system with 3 Enhanced Portfolio radios, 1 Enhanced Portfolio control station, and 3 groups. Note that radios 1, 2, and 50 comprise group 1; radios 1, 3, and 50 comprise group 2; and radios 2, 3, and 50 comprise group 3. Radio 50 is a control station with an attached PC with USB (e.g. a dispatch op position) and an attached Bluetooth device. Radio 50 is a member of all 3 groups.

The following explanation provides an understanding of how these four networks coexist and how the Enhanced Portfolio radio, its attached PC with USB, and its attached Bluetooth device deal with these multiple points of presence. On the CAI, it is assumed that every radio, every attached PC with USB, every attached Bluetooth device, and every group must have a uniquely routable IP address. Since a layer 2 Id (theoretically up to 2^{24} layer 2 Id) must maps to a unique IP address, the network supporting the radios is a class A network space supporting single destination endpoint IP datagrams (i.e. unicast). This network is referred to as the “CAI Subscriber Network, 12.x.x.x”; although the NetID is actually provisioned by CPS (range is 1 to 126). Since every attached PC with USB must have a unique, IP address (at most, up to one attached PC per radio), the network supporting the attached PCs is a class A network space supporting single destination endpoint IP datagrams (i.e., unicast). This network is referred to as the “CAI Attached PC Network, 13.x.x.x”, although the NetID is actually derived from the provisioned CAI Subscriber Network NetId.

Likewise, since every connected Bluetooth device must also have a unique, IP address (at most, up to one connected BT device per radio), the network supporting the PCs attached with USB is also a class A network space supporting single destination endpoint IP datagrams (i.e. unicast). This network is referred to as the “CAI Attached BT Network, 14.x.x.x”. The NetID is derived from the provisioned CAI Subscriber Network NetID.

Finally, since every group (theoretically up to 2^{24} talk groups may be present on a system) must have a unique IP address, the network supporting the talkgroups is a class D network space supporting multiple destination endpoint IP datagrams (i.e., multicast). This network is referred to as the “CAI Talkgroup Network, 225.x.x.x”, although the NetID is actually provisioned by CPS (range is 224 to 239). Note that even though the Class D multicast address space is used on the CAI, it is not necessary for the radio to implement multicast or IGMP.

The CAI Radio Network NetID is provisioned by CPS while the CAI Attached PC with USB NetID is automatically set to be the CAI Radio Network NetID +1, and the CAI attached Bluetooth NetID is automatically set to be the CAI Radio Network NetID + 2, being mindful of the boundary condition (e.g., when the attached PC with USB Network NetID = 126, the Subscriber Network NetID + 1 = 1). The CAI Talkgroup Network NetID is separately provisioned by CPS.

Radios may receive IP datagrams from the CAI Radio Network (e.g., 12.x.x.x), the CAI Attached PC with USB Network (e.g., 13.x.x.x), the CAI Attached Bluetooth Network (e.g. 14.x.x.x) or the CAI Talkgroup Network (e.g., 225.x.x.x) when the destination IP address matches the list of IP addresses (derived from the radio's layer 2 addresses of interest) as specifically provisioned in their configuration data via CPS. Radios always transmit IP datagrams using their CAI Subscriber Network presence (e.g., 12.x.x.x) as the source IP address independent of which CAI IP network space they are transmitting into. In other words, a group IP address homed on the CAI Talkgroup Network (225.x.x.x) can never be used as a source IP address, since this would indicate that a group is sourcing the IP datagram, which is clearly incorrect since an individual subscriber is responsible for generating an IP datagram.

4.2 Private Radio Network

In addition to having a presence on the CAI Radio Network, the CAI Attached PC with USB Network, and the CAI Talkgroup Network, the radio also has a presence on a local, Private Radio Network (e.g., 192.168.10.x). Minimally, the radio itself (e.g., for internal applications such as CPS and tuner) has a statically assigned IP address on the Private Radio Network. Optionally, other devices, such as an PC attached with USB or Bluetooth PANU device, have a presence on the Private Radio Network. The radio dynamically assigns an IP address to the PC attached with USB through standard DHCP procedures (Radio acts as DHCP Server.). The Private Radio Network's network ID and subnet mask are CPS provisionable and attached devices obtain this information via standard DHCP procedures. Additionally, the radio's host ID is CPS provisionable and attached devices learn the radio's full IP address when they receive a response from the radio's DHCP Server (i.e., by examining the source IP address).

Since the Private Radio Network is managed locally, by the radio, a NAT is required to allow all of the devices on the Private Radio Network and the subscriber's internal applications to share the network interfaces on the CAI side of the radio.

The radio subnet with NAT concept using attached PCs with USB is illustrated in Figure 12.

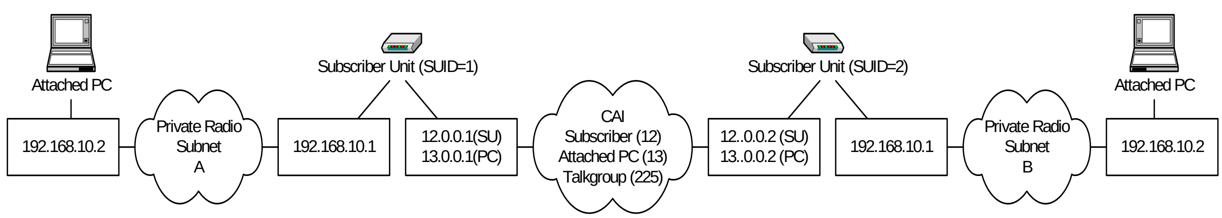

Figure 12 – Radio Subnet with NAT (Attached PCs With USB)

MOTOTRBO core radios use a custom IP stack, where a Motorola-proprietary IP Addressing mechanism is used. The PC application uses the subscriber CAI IP address (e.g. 12.x.x.x) to access the remote radios. For local radios, the PC can either use the private radio network IP address 192.168.10.1 when connected with USB, or the subscriber CAI IP address.

Refer to **Figure 4** for a summary of the corresponding CPS settings regarding the above example. The CPS "Radio IP" setting (192.168.10.1) corresponds to the radio's IP address at the private radio network based on the USB connection. The CPS "Accessory IP" setting (192.168.10.2) corresponds to the PC's IP address at the private radio network based on the USB connection.

MOTOTRBO Enhanced Portfolio and SL Series radios use a standard Linux IP Stack, which follows the standard IP stack behavior and is used across multiple products and platforms. The PC application uses the subscriber CAI IP address (e.g. 12.x.x.x) to access the remote radios. The PC cannot use the subscriber CAI IP address for local radios, and uses either:

- The private radio network IP address 192.168.10.2 when the PC is connected with USB, or
- The private radio network IP address 192.168.11.2 when the PC is connected via Bluetooth PAN.

The radio subnet with NAT concept using attached Bluetooth devices is illustrated in **Figure 13**. For local radios using Bluetooth connections the source IP address must be the local Bluetooth network IP address (e.g. 192.168.11.1)

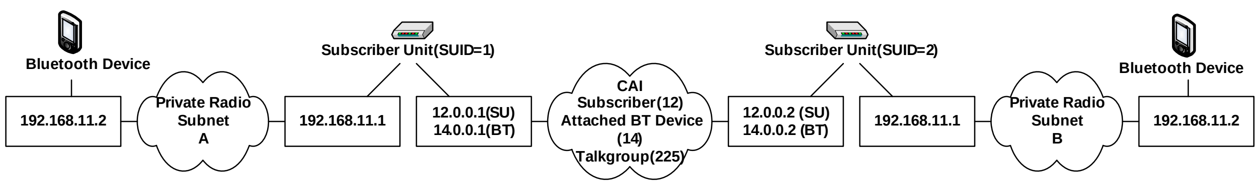

Figure 13 – Radio Subnet with NAT (Attached Bluetooth Devices)

The local USB network IP address can be programmed in CPS for each radio. The Bluetooth IP address is derived from the USB network IP address configuration.

Refer to **Figure 4** for a summary of the corresponding CPS settings regarding the above example. The CPS "Bluetooth IP" setting (192.168.11.1) corresponds to the radio's IP address at the private radio network based on the Bluetooth PAN conection. The CPS "Bluetooth Accessory IP" setting (192.168.11.2) corresponds to the Bluetooth device's IP address at the private radio connection based on the Bluetooth PAN connection.

The NAT address translation is positioned in the radio between the CAI network interface and the IP routing function. This means that all devices located on the local, Private Radio Network may directly exchange IP datagrams without traversing the NAT. When a device located on the Private Radio Network desires to send an IP datagram beyond the radio (e.g., from 192.168.10.x to 12.x.x.x, 13.x.x.x, 14.x.x.x, or 225.x.x.x), the NAT is traversed to replace the source IP address with a routable address on the CAI interface (e.g., 12.x.x.x). Likewise, when a radio receives an IP datagram on the CAI interface, it traverses the NAT to translate the destination IP address into a routable address on the Private Radio Network (e.g., 192.168.10.x). In addition to providing IP translation, the NAT may additionally provide UDP port translations. The following discussion is intended to help clarify this operation through use of an example.

Figure 14 depicts the protocol stack in a radio using a USB connection to the local device.

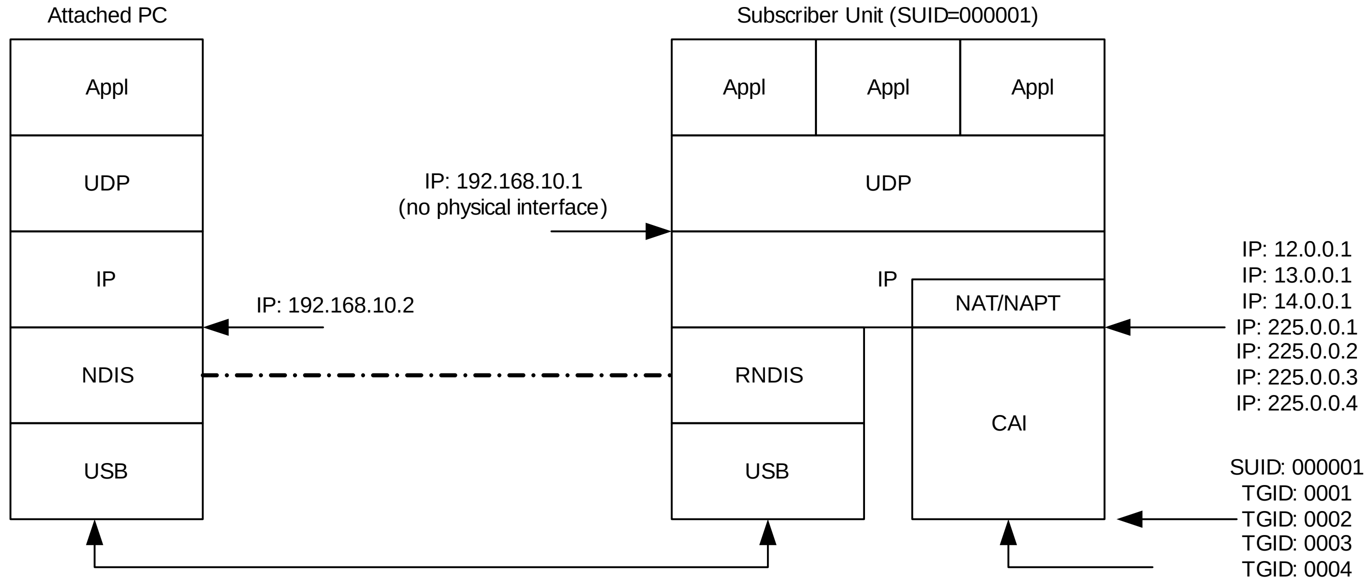

Figure 14 – A Radio’s Protocol Stack for Data (USB Connection)

On the right is the CAI interface, which filters the link layer addresses SUID 000001, TGID 0001, TGID 0002, TGID 0003, and TGID 0004 and allows data PDUs containing these destination addresses to pass up to network layer. At network layer the destination addresses appear as 12.0.0.1, 13.0.0.1, 14.0.0.1 225.0.0.1, 225.0.0.2, 225.0.0.3, and 225.0.0.4. These are subjected to the NAT and translated to a 192.168.10.x destination address before routing on the radio is attempted.

NOTE

Capacity Plus supports 8 bit talkgroups 1-254; 255 is reserved.

On the left is the USB interface that connects to an attached PC, which is assigned the layer 3 address 192.168.10.2. Finally, note that the applications within the radio all share the Private Radio Network IP address 192.168.10.1.

The protocol stack for a radio serving as a Bluetooth PAN-NAP is almost identical to that of a radio using the USB connection. The IP addresses used to connect to and identify a PANU device are slightly different, and are based on the “Radio IP” address configured in CPS. Refer to **Figure 15**.

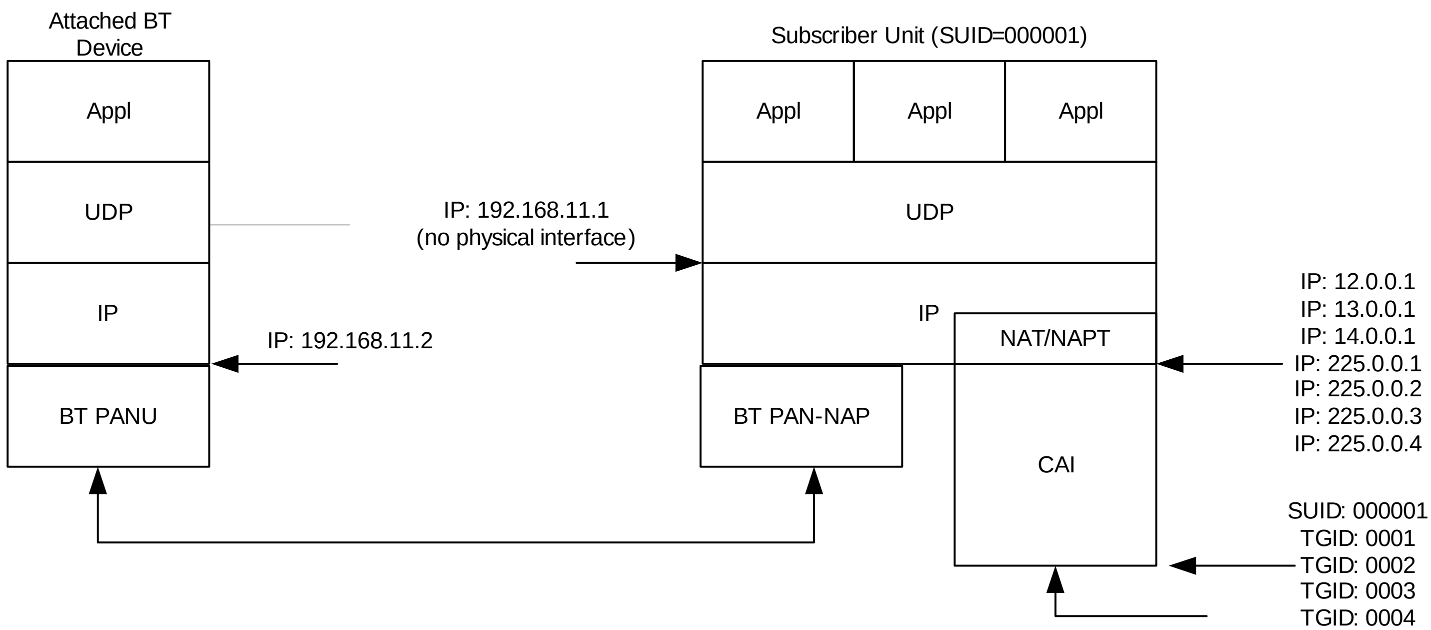

Figure 15 - A Radio's Protocol Stack for Data (BT PAN Connection)

Since it may not be intuitively obvious why a private radio network and a NAT are necessary, a brief justification is in order. Firstly, the applications running on the attached PC with USB or Bluetooth device may be interested in receiving datagrams destined to talkgroups (e.g., the 225.x.x.x network space) in addition to its own unicast datagrams (e.g., the 13.x.x.x or 14.x.x.x network space). This would either require multicast IP (IGMP) or multiple IP addresses (e.g., one per talkgroup) to be implemented by the radio, which is complex. Secondly, some talkgroup datagrams will be destined to internal applications while others are destined to the PC attached with USB and a means of selectively routing these IP datagrams is needed. Thirdly, some unicast datagrams destined to the radio (e.g., the 12.x.x.x space) may need to be preferentially routed to the attached PC, as is the case when the PC-based text message application is being used at a particular radio. Finally, the NAT allows mitigation of IP network space conflicts between the CAI networks and any Customer or commercial Internet network space that the PC attached with USB may be connected to (directly or indirectly). The Private Radio Network combined with NAT appears to address each of these needs.

The core MOTOTRBO™ system allows a customer to select between two NAT configurations: "Default Configuration" and "Forward to PC Configuration with USB Configuration". The Enhanced Portfolio & SL Series MOTOTRBO™ system allows a customer to select between three NAT configurations: "Default Configuration", "Forward to PC with USB Configuration" and "Forward to PC with Bluetooth".

The configurations are selectable via the "Radio Network" settings in CPS. Refer to **Figure 16**Error! Reference source not found..

- "Default Configuration" is selected in CPS as "disabled".

- “Forward to PC with USB Configuration” is selected in CPS as “via USB”
- “Forward to PC with Bluetooth Configuration” is selected in CPS as “via Bluetooth”

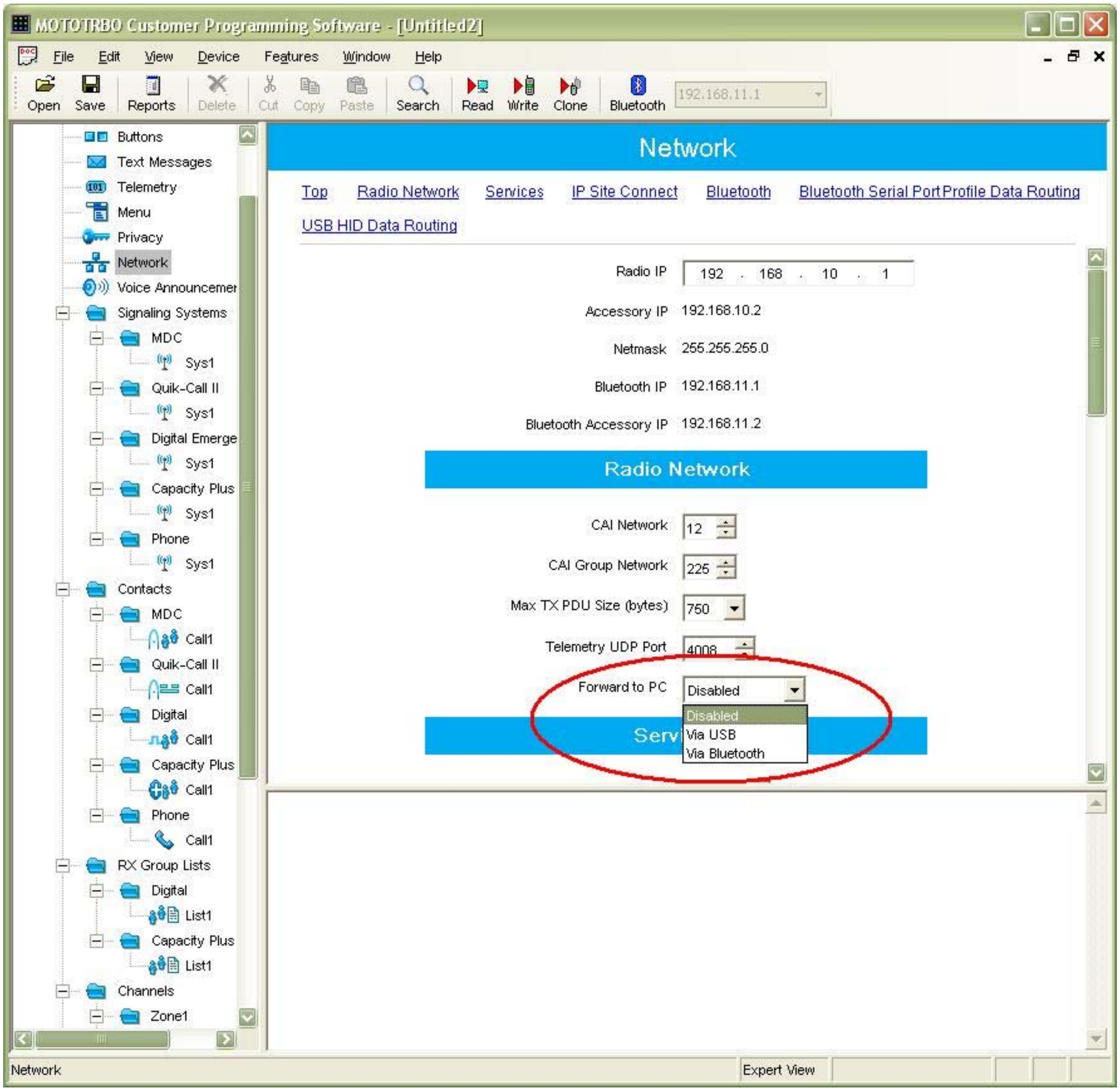

Figure 16: "Forward to PC" Settings (Enhanced Portfolio & SL Series Radios)

In the case of “Default Configuration”

- All the datagrams address to a radio are sent to the internal applications (i.e. applications in a radio).
- All the datagrams address to an IP-capable peripheral (e.g. a PC connected either with USB or Bluetooth PAN) are sent to external applications (i.e. applications in a PC).

- All the datagrams address to a talkgroup are sent to the applications internal to a radio.

In case of “Forward to PC Configuration with USB Connection”

- All the datagrams address to a radio are sent to the internal applications (i.e. applications in a radio), except for Text Messaging data (UDP port #4007) and third party application raw data. Text messaging data and third party application raw data are forwarded to the PC connected with USB.
- All the datagrams addressed to a USB IP-capable peripheral (e.g. a PC) are sent to the external applications connected with USB.
- All the datagrams addressed to a Bluetooth PAN IP capable peripheral (e.g. a PC) are sent to the external applications connected with Bluetooth PAN.
- All the datagrams addressed to a talkgroup are sent to the external applications connected with USB except that the destination port is for Option Board or non-IP peripheral proxy ports.

When the radio tries to forward the message over a severed USB connection, the radio discards the message and sends an ICMP message to the source radio.

In case of “Forward to PC Configuration” with Bluetooth PAN Connection”

- All the datagrams address to a radio are sent to the internal applications (i.e. applications in a radio), except for Text Messaging data (UDP port #4007) and third party application raw data. Text messaging data and third party application raw data are forwarded to the PC connected with Bluetooth PAN.
- All the datagrams addressed to a USB IP-capable peripheral (e.g. a PC) are sent to the external applications connected with Bluetooth PAN.
- All the datagrams address to a talkgroup are sent ot the external applications connected with Bluetooth PAN except that the destination port is for Option Board or non-IP peripheral proxy ports.

When the radio tries to forward the message over a severed Bluetooth PAN connection, the radio discards the message and sends an ICMP message to the source radio.

Section 5

5.0 Option Board and Non-IP Capable Peripheral interface

MOTOTRBO™ provides an interface for a 3rd party developer to add an embedded device, an Option Board, to expand the capability of a radio. The Option Board communicates to the main board of the radio. The physical layer interface between the Option Board and the main board is the Synchronous Serial Interface (SSI), a multi-slotted time division multiplexed (TDM) communication channel. Please see reference [[3]] for more information.

Aside from the Option Board interface, the MOTOTRBO™ external accessory connector (UC, MMP and MAP) provides a mechanism for a third party to add a Non-IP Capable Peripheral to expand the capability of the radio. The Non-IP Capable Peripheral is able to physically connect to the radio via the radio's accessory port and operates as a USB host in the connection. Please see reference [16] for more detail on the design and development of a MOTOTRBO™ compatible Non-IP Capable Peripheral.

Both the Option Board and the Non-IP Capable Peripheral have no IP stack and use XCMP commands to communicate with the radio. MOTOTRBO™ provides four UDP ports for Option Board and Non-IP Capable Peripheral, respectively, to route received data from the UDP/IP Network to the Option Board/Non-IP Capable Peripheral. So the Option Board and the Non-IP Capable Peripheral can still send text messages, telemetry, location data, and raw data to the IP network even though they are non-IP capable devices. The UDP proxy port definition for Option Board and Non-IP Capable Peripheral are shown in the table below.

XCMP Protocol Type	Protocol Name	Option Board UDP Proxy Port	Non-IP Capable Peripheral UDP Proxy Port
0x00	Location Request Response Protocol	4063	4068
0x20	Text Messaging Protocol	4061	4066
0x30	Telemetry Protocol	4062	4067
0x50	Raw Data	4004	4069

The MOTOTRBO™ radio provides a proxy application that sends and receives a datagram on behalf of the Option Board applications and Non-IP Capable Peripheral applications. The Option Board (or Non-IP Capable Peripheral) sends a datagram by packing the payload of the datagram into XCMP's "Data Session Request" or "Data Session Broadcast" messages, which allows the sending application in the Option board to specify the destination IP address, destination UDP port, and the source UDP port. An application in an Option Board specifies the protocol type (Location, Text Messaging, Telemetry, and others). The proxy application maps the protocol type to a corresponding source port number as shown in the table above. Please see reference [4] and [5] for more information about the XCMP data session command.

Figure 17 is an example that a Non-IP Capable Peripheral sends a Text Message to a remote radio.

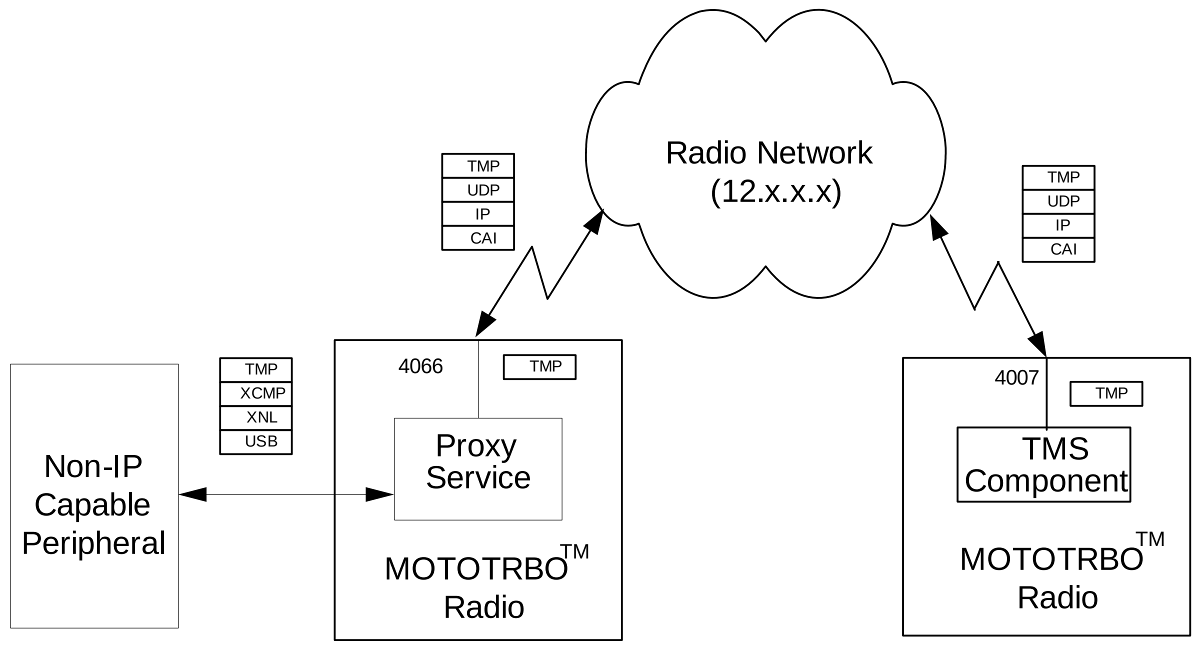

Figure 17 – An Example of Non-IP Capable Peripheral Communication

When the Non-IP Capable Peripheral wants to send a text message to a remote radio, it sends the text message to the local radio with the destination IP address set to the remote radio. The destination UDP Port is set to the text message port and the protocol type set to Text Message (0x20) in the XCMP Data Session message. The local radio forwards the text message as an UDP payload over the air to the remote radio with the source UDP port set to 4066, and the source IP address set to the radio's CAI radio network IP address (i.e.12.0.0.1).

When the remote radio sends back a text message to the Non-IP Capable Peripheral, it sends the text message in the UDP message with the destination IP address set to the target radio and the destination UDP port set to the Non-IP Capable Text Message proxy port (4066). When the target radio receives the text message on the Non-IP Capable Text Message proxy port (4066), it forwards the text message to the Non-IP Capable Peripheral using the XCMP data session message. The source IP address, the source port number, and the destination port number from the datagram are mapped into the “Destination Address”, “Destination Port”, and “Data Protocol & version” fields of the “Data Session Request” message.

Please note that the communication between the Non-IP Capable Peripheral (or Option Board) and the remote radio can only happen while the radio is in digital RF mode. The XCMP data session command does not guarantee the data transmission between the Non-IP Capable Peripheral (or Option Board) and the remote device, and it does not send back any response to the sender when it fails to send out the data to the target device. So, for the peripheral application's delivery and retry mechanism, it is recommended that the application query the current channel to determine if the radio is in digital RF mode before sending any data over the air.

Before MOTOTRBO R1.7, the third party devices (e.g. Option Board, Non-IP Capable Peripheral or PC attached to the radio) originated data to an application server can not be sent on the Data Revert channel in the MOTOTRBO Capacity Plus system or the GPS revert channel in the MOTOTRBO conventional system as the radio does.

Since MOTOTRBO R1.7, in the Capacity Plus system when the third party devices originated data, such as Text Message, Location, ARS, Telemetry or Raw Data message, is targeted to a data server, it can be transmitted on the Data Revert channel; in the MOTOTRBO Conventional system, when the third party device originated location data is targeted to a location server, it can be transmitted on the GPS Revert channel. The data server's IP address must be pre-configured in the radio, e.g. Text Message Radio ID, ARS Radio ID, or be saved when the radio receives the location request from the location server. When the data is not targeted to a data server, e.g. another radio, or a PC which is different from the data server IP addresses, it is sent through the trucked channel in the Capacity Plus system or the selected channel in the Conventional system.

Since MOTOTRBO R1.8A, when the third party device originated location data is targeted to PC attached to the radio, it can be sent on the Data Revert channel in the MOTOTRBO Capacity Plus system or the GPS revert channel in the MOTOTRBO conventional system.

5.1 Multi Channel Routing for External Data Applications

In case of wide coverage area or in case of high-density system, a single dispatcher-based application such as PC-based text messaging and Location Tracking may be required to monitor multiple set of SUs (e.g. multiple fleets of taxies). The channel of a radio depends upon its current personality and the personality is selected by the user of the radio. This makes the relationship between a radio and its channel dynamic. To reduce the complexity of applications and to keep them uniform over multiple platforms, MOTOROLA SOLUTIONS designed a Multi-Channel Device Driver (MCDD) for MOTOTRBO to keep the applications transparent of any change in the current channel of a radio. It provides a solution for communication between a data application in a dispatcher's PC and radios on up to four logical channels. This implies that an application on a dispatcher's PC is able to send and receive IP datagrams to a radio on any of the four logical channels. The UDP/IP addresses of the radio and the applications as seen by each other should remain same. Please see reference [14] for more detail on the MCDD configuration.

NOTE

The Multi-Channel Device Driver (MCDD) is required for a conventional and IP Site Connect system for the preprocessing of the ARS.

The MCDD is not compatible with Capacity Plus. In Capacity Plus the association of a radio to a channel is no longer needed because the radio transmits and receives the IP datagrams on any open channel.

5.2 Performance

MOTOTRBO™ provides an IP based data pipe that is shared by multiple applications such as Location, Text Messaging, and Telemetry. Since the data pipe is a common resource, a data application must use it in a disciplined way.

This section provides a set of guidelines for a data application. If followed by all applications, the guidelines improve the fair sharing of the data pipe and reduce the probability of buffer overflow in the radios or Control Stations. It does not guarantee that the data messages will not be lost.

More specifically, this section provides a process to calculate three parameters that constrains the way a data application uses the data pipe. The three parameters are “instantaneous message transmission rate”, “average message transmission rate”, and “retry rate”.

<u>Message Flight Time</u>:
The first step is to calculate the serialization time of a typical message. Let ‘p’ is the size of the application layer payload in octets. At network layer the size of the payload is “p + 8 + 20” octets where 8 is the size of the UDP header and 20 is the size of the IP header. The number of data blocks required for this payload for at link layer (confirmed packets) is given by the expression “Ceiling [((p + 28) - 12) ÷16] + 1”, where 16 is the number of user bytes in a block and 12 is the number of user bytes in the last block. The expression can be simplified to “Ceiling [p ÷16] + 2”. To the data blocks, the following additional blocks are added before transmission:

One block long Data Header,

- One block (in repeater mode) and two blocks (in direct mode) long preamble for battery power saving
- Two blocks (in repeater mode) and one block (in direct mode) long Terminator LC
- Two empty bursts (in repeater mode) and approx. one burst (in direct mode) for channel access

Thus, in worst case conditions, p octets long application layer payload requires “Ceiling [p ÷16] + 8” bursts for transmission i.e. “(Ceiling [p ÷16] + 8) * 60” milli seconds.

<u>Instantaneous Message Transmission Rate</u>:

A radio or a Control Station has a buffer for 10 data messages. These buffers are intended to take care of channel access time and messages from other applications. An application should avoid using these buffers for buffering more than one of its messages and therefore should delay the second message by the Message Flight Time of the first message.
Thus the Instantaneous Message Transmission Rate of an application should be less than one message per Message Flight Time. This implies that the minimum interval between two consecutive messages should be the Message Flight Time.

Average Message Transmission Rate:
The inverse of the Message Flight Time gives the upper limit on the maximum transmission rate. Due to slotted Aloha nature of air interface, the maximum transmission rate is limited to approximately 35% of the upper limit. Since the air interface is a shared resource between voice and other data applications, an application should use, on the average, only a fraction of the 35%.

Retry Rate:
In worst condition, at link layer, a confirmed message is sent 3 times by a control station. Between each retry, a sender waits for the layer 2 acknowledgement. The default retry time is 3.75 sec. Thus it may take (3 * Message Flight Time + 2 * Retry Time) seconds for the message to reach its destination radio in direct mode and 60 msec more in repeater mode. This time encompasses the layer 2 acknowledgement.

The same is true for the application layer acknowledgement from the radio to arrive at the control station. Since protocol acknowledgements are generally smaller than the payload messages, the Message Flight Time for the ACK can be calculated separately. Assuming negligible delay between a control station and the PC, an application should wait for another (3 * ACK Message Flight Time + 2 * Retry Time) seconds for the application layer ACK message to reach the control station.

Therefore the total time an application should wait before sending a retry message is (3 * Message Flight Time) + (3 * ACK Message Flight Time) + (4 * Retry Time) + processing time for acknowledgement in radio + 60 msec (in case of Repeater mode)

5.3 External Data Applications in IP Site Connect System

The IP Site Connect feature became available on the MOTOTRBO Repeater as of System Release 1.4. This feature enables users to link up to 15 MOTOTRBO repeaters, or 30 MOTOTRBO repeaters with one Wide Area Channel each, via an IPv4 network connection.

By linking the repeaters together, the IP Site Connect feature provides the following advancements to the radio system:

- Expands communication to multiple locations simultaneously
- Permits contiguous or non-contiguous wide area coverage

- Enhances the single site capacity
- Seamless roaming from one coverage area to another without manual intervention or interruption in contiguous coverage area systems

When the MOTOTRBO repeater operates in the IP Site Connect mode, it supports two TDMA time slots or two logical channels. They are a Local Area Channel to repeat the on-site traffic or a Wide Area Channel to send the traffic to the peers while the call is transmitted at the sites.

In an IP Site Connect System, a third-party data application connects to one or more Control Stations. Each Control Station is dedicated to send or receive traffic on a logical channel, which can be a Local Area Channel or a Wide Area Channel. There is no change required to modify a third party data application to be IP Site Connect compatible.

A third party application can be a centralized server for multiple IP Site Connect Systems when the Control Stations are set to the Wide Area Channels of the IP Site Connect Systems. MCDD is required when two or more Control Stations are connected to the third party application.

When sending the location data to the server, the MOTOTRBO radio uses unconfirmed data transmission in a Wide Area Channel, and confirmed data transmission in a Local Area Channel.

For any other data message, no matter which channel it is in, the MOTOTRBO radio uses confirmed data transmission for individual data message, and unconfirmed data transmission for group data message.

891

6880309T36