MASS SOFTWARE UPGRADE PROTOCOL SPECIFICATION

Version 3.0
April 2012

Roles	Function	Name
Authors	Associate Lead – Software, AOCI	Chetan Ethapay
Reviewers	Systems Architect, North Andover	David Kopp
	Product Development Lead, North Andover	Ron Naismith
	Technical Lead – Software, AOCI	Pawan Modi
	Associate Lead – Software, AOCI	Shrikant Pawar
	Technical Lead – Software, AOCI	Rajendra Alluri
	Senior Engineer – Software, AOCI	Deepak Yadav
Approvers	Director, AOCI	Ramesh Phatak

Document Revision History			
Version	**Date (yyyy/mm/dd)**	**Authors**	**Modifications Details**
1.0.1	2011/03/29	Chetan Ethapay	Initial Creation
1.0.2	2011/06/17	Chetan Ethapay	Changed the format of the table headers in all the tables
1.0.3	2011/08/18	Chetan Ethapay	Added “data length” field in the upgrade message
1.0.4	2011/11/25	Chetan Ethapay	Added “MSU completed message”
2.0	2011/12/12	Chetan Ethapay	Added discovery messages “Who-Is” and “I-Am”, added “Status Message” from the clients
2.1	2012/02/29	Chetan Ethapay	Added abbreviations table, “Group Creation” and “Group Creation Acknowledge” messages, added “CCM completed” message and “Transfer Aborted” message
3.0	2012/04/25	Chetan Ethapay	Added SCM transfer completed message and description, added device parameter field in the I-Am message

Abstract

Mass Software Upgrade [MSU] is a reliable network protocol for carrying out simultaneous firmware/software upgrade process of a large number of systems. It uses a combination of multicast and unicast datagram protocols for data transfer. Reliability is achieved by selective re-transmission of the desired portion of the data upon request from the systems participating in the upgrade process.

1. Abbreviations

Abbreviation	Expansion
MSU	Mass Software Upgrade
UDP	User Datagram Protocol
IGMP	Internet Group Management Protocol
FA	Firmware Archive
CN	Chunk Number
SN	Sequence Number
CCM	Chunk Complain Mode
SCM	Sequence Complain Mode
CM	Complain Mode
GM	Group Message
IM	Individual Message

2. Introduction & Overview

The objectives of Mass Software Upgrade process are: 1) to simultaneously upgrade firmware/software on many systems, 2) to encourage indirect or implicit (via programs) use of remote computers, 3) to transfer data reliably and efficiently even on high traffic network. 4) Automatic recovery of upgrade process even in cases of network re-establishment.

This paper assumes knowledge of the User Datagram Protocol (UDP), Unicast and Multicast (IGMP) data transmission over the Ethernet.

3. Terminology

a. Mass Software Upgrade [MSU] – The process of carrying out simultaneous upgrade process of a large number of clients which require the same set of firmware package and/or configuration files.

b. Firmware Archive [FA] – Refers to a single binary file or a number of firmware files archived to form a single package. The files mentioned here may also refer to configuration files containing plain text. The terms FA and File are used interchangeably in this document.

c. Server System(s) – The system(s) which is the source of the firmware archive. Note: Although a number of server systems can be present on the same network, at a particular instant of MSU process, only one system is expected to be participating. The term server used in this document refers to the Server System.

d. Client System(s) – The system(s) in which the firmware needs to be updated. Note: The term client(s) used in this document refers to the Client System(s).

e. Chunk – Binary file or firmware archive broken down into a number of pieces of equal and predefined size. Note: The last chunk of the file may be of a different length when compared to the other chunks. Each chunk is numbered incrementally starting from one. This number is referred to as the chunk number [CN].

f. Sequence Number [SN] – Each chunk is further divided into a predefined number of units referred to with numbers starting from one.

g. Complain Mode – Selective re-transmission of a part of the firmware archive.

4. Summary of Operation

MSU process uses a combination of Multicast and Unicast datagram protocol for communication over the Ethernet for data transfer. It involves reliable data transfer using a mode of operation referred to as the complain mode for selective re-transmission of a part of the firmware archive.

Figure 1 shows an example setup for MSU process.

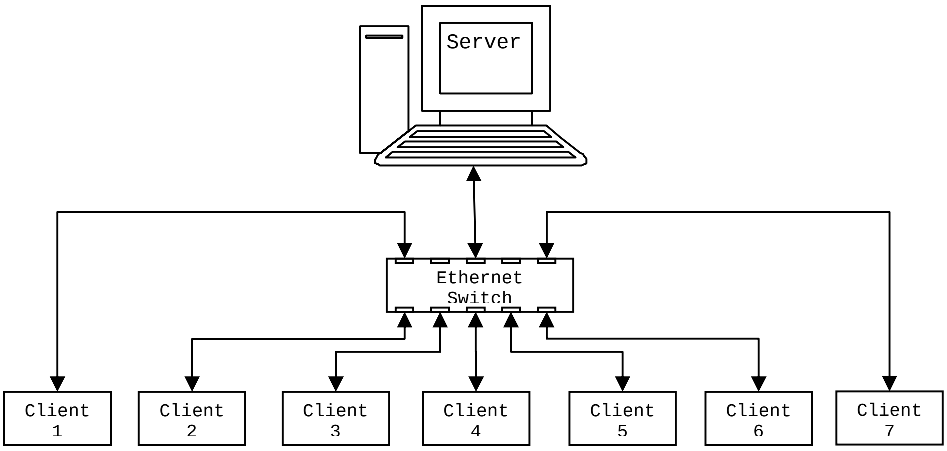


Figure 1 - Example of MSU Process Setup

5. Design Goals & Constraints

MSU process has been designed to make use of the advantages of multicast datagram protocol to upgrade systems on a large scale simultaneously in order to save time and increase the network's efficiency to a large extent.

6. Reliability

Reliable data delivery over the unreliable network is achieved by using the complain mode in the MSU process. The MSU process enters into complain mode after the transfer of each of the chunks hereby referred to as sequence complain mode [SCM] for re-transmission of a selected part of the chunk indicated by the SN. By the end of the complete FA transfer, the MSU process enters into complain mode for re-transmission of the selected chunk indicated by the CN. This process is hereby referred to as the chunk complain mode [CCM].

7. Precondition

1. Each of the client systems must have a unique MAC address.
2. Each of the client systems must have unique IP address.

8. Protocol Data Format

All the data in the MSU protocol definitions are represented in the **Big-Endian** format.

9. Device Discovery Messages

The server sends the '**Who-Is**' multicast message on the default multicast IP address and port number to discover all the clients on the network. The clients reply with the **'I-Am'** unicast message.

a. Who-Is

Byte-0	Byte-1	Byte-2	Byte-3
Opcode (4) | Subcode (4)	IsRange	Reserved	IP Version (1) | Protocol Version (3) | reserved (4)
Start Range/DeviceID			
End Range			

Table 1 - Who-Is Discovery Message Format (GM)

🟧	Optional Field
▨	Fixed Length Field
🟦	Variable Length Field

- Opcode - MSU_DEVICE_DISCOVER (2)
- Subcode – WHO_IS (1)

- IsRange - TRUE (1) or FALSE (0). If TRUE, the Start Range and End Range should be provided. If FALSE, Start Range is interpreted as the Device ID and the End Range is filled with zeros/null character.
- IP Version – IPV4 (0)

  IPV6 (1) – reserved
- Protocol Version (3) – This field should be set to 2.
- Start Range – Starting IP address in the discovery process. This is applicable if IsRange is TRUE.
- Device ID – IP address of the device, applicable if IsRange is FALSE.
- End Range – End IP address in the discovery process. This is applicable if IsRange is TRUE.

b. I-Am

Byte-0	Byte-1	Byte-2	Byte-3
Opcode (4) | Subcode (4)	GroupID	Reserved	IP Version (1) | Protocol Version (3) | reserved (4)
DeviceID			
T/F | HW_ID_Len	T/F | Product_ID_Len	T/F | Product_Name_Len	T/F | Model_Name_Len
T/F | Vendor_ID_Len	T/F | FW_SW_Ver_Len	T/F | Major_Minor_Revison_Len	T/F | Device_Loc_Len
T/F | MSU_Comm_Param_Len	T/F | Dev_Param_Len	Reserved	
Reserved for user specific field			
Variable Length Data			

Table 2 - I-Am Discovery Message Format (IM)

- Opcode - MSU_DEVICE_DISCOVER (2)
- Subcode – I_AM (2)
- GroupID – Group number to which this device belongs. The GroupID is supplied by the server during the Group Creation process. If the device is not yet assigned to any Group, by default, the GroupID is set to zero.
- IP Version – Refer to Table 1 description.
- Protocol Version (3) – Refer to Table 1 description.
- DeviceID – Unique representation of the device such as the IP address for Ethernet devices.
- T/F – Bit to suggest if the field exists or not.
- HW_ID_Len – Length of the Hardware Identification string.
- Product_ID_Len – Length of the Product Identification string.
- Product_Name_Len – Length of the Product Name string.
- Model_Name_Len – Length of the Model Name string.
- Vendor_ID_Len – Length of the Vendor Identification string.
- FW_SW_Ver_Len – Length of the Firmware Version string.
- Major_Minor_Revison_Len – Length of the Major and Minor Revision string.
- Device_Loc_Len – Length of the Device Location string.
- MSU_Comm_Param_Len – Length of the MSU Communication Parameter string.
- Dev_Param_Len – Length of the Device Parameter field. This field is as shown in
- Table 3. The server, having collected the information from all the clients has to use the largest of all the values in the corresponding fields. This is in order to support even the slowest of all the clients.

Name	Size in Bytes	Units	Description
Sequence_Delay	2	milliseconds	Delay between consecutive data packets
SCM_Delay	2	milliseconds	Delay between consecutive SCMs
CCM_Delay	2	milliseconds	Delay between consecutive CCMs
CCM_Retry	1	number (0, 1,…255)	Number of CCM retries
SCM_Retry	1	number (0, 1,…255)	Number of SCM retries
Timeout_Val	1	seconds	Timeout value between any two consecutive response from the MSU server

Table 3 - Device Parameter Field

- User specific Data. – User can define own identity to recognize the device. User can append the data at the end of the defined variable length data.
- Variable Length Data – String of variable length, the length is specified by the previous field. For example, File Name Length field specifies the length of this field.

10. Group Creation

The server sends Group Creation unicast message to the individual clients with the GroupID information.

Byte-0	Byte-1	Byte-2	Byte-3
Opcode (4) | Subcode (4)	GroupID	Reserved	IP Version (1) | Protocol Version (3) | reserved (4)

Table 4 – Group Creation (IM)

- Opcode – MSU_DEVICE_GROUP (3)
- Subcode – SET_GROUP_ID (1)

  RESET_GROUP_ID (2)
- GroupID – Group number is specified by the server in order to group the devices. By default, the GroupID is zero. Using the RESET_GROUP_ID Subcode, the server can reset the GroupID to zero.
- IP Version – Refer to Table 1 description.
- Protocol Version (3) – Refer to Table 1 description.

11. Group Creation Acknowledge

Client Systems need to send ACK message to the server once they handle the group create message from server. If server does not receive the ACK message from the devices, it will resend the group create message. Number of retries is subjected to the implementation.

Byte-0	Byte-1	Byte-2	Byte-3
Opcode (4) | Subcode (4)	GroupID	Reserved	IP Version (1) | Protocol Version (3) | reserved (4)

Table 5 – Group Creation Acknowledge (IM)

- Opcode – MSU_DEVICE_GROUP (3)
- Subcode – CREATE_GROUP_ACK (3)
- GroupID – device Group id (optional).
- IP Version – Refer to Table 1 description.
- Protocol Version (3) – Refer to Table 1 description.

12. Update Process

Server initiates the MSU process by multicasting notification message. The default multicast IP address, port number, the number of times this message is sent and the delay between each of the notification messages are implementation specific. The Notification message packet contents are shown in Table 6.

Byte-0	Byte-1	Byte-2	Byte-3
Opcode (4) | Subcode (4)	File Number		IP Version (1) | Protocol Version (3) | reserved (4)
File Size (in bytes)			
Number of Chunks			
Sequence Number Limit		Sequence Size Limit (in bytes)	
Server IP Address			
Multicast Address			
CM Multicast Address			
Port Number		CM Port Number	
TransactionID			
File CRC			
T/F | File_Name_Len	T/F | Dest_Path_Name_Len	T/F | GroupId	Update Timeout
Variable Length Data			

Table 6 – Notification Message Format (GM)

- Opcode – MSU_DEVICE_UPDATE (1)
- Subcode – UPGRADE (1)

  DOWNGRADE (2)

  FORCE_UPGRADE (3)
- File Number – In case of upgrade process involving multiple files, this number represents the index of the file that is being transferred. This number starts from 1.
- File Size – Size of the file in bytes that is being transferred.
- Number of Chunks – Total number of chunks in the current file transfer.
- Sequence Number Limit – Maximum number of sequences in the first chunk up to the last but one chunk. Note that the last chunk may have a smaller number of sequences depending on the file size. This limit should not be more than 32.
- Sequence Size Limit – Maximum size in bytes of the data payload of a single packet sequence. Note that the last sequence of the last chunk may have a smaller size depending on the file size.

- Server IP Address – IP address of the Server System. This address is used by the clients to send unicast messages to the server.
- Multicast Address – The client systems that are interested in participating in the MSU process need to join this multicast address.
- TransactionID – Randomly generated number which uniquely represents one MSU cycle.
- Port Number – The client systems that are interested in participating in the MSU process need to listen on this port to receive multicast messages.
- CM Multicast Address – The client systems which are interested in participating in the CM either in SCM or CCM need to join this multicast address.
- CM Port Number – The client systems which are interested in participating in the CM either in SCM or CCM need to listen on this port to receive CM multicast messages.
- File CRC – 32-bit CRC of the file.
- T/F – Bit to suggest if the field exists or not.

  1 – Field exists.

  0 – Field does not exist.
- File Name Len. – Length of the file name.
- Dest. Path Name Len. – Length of the destination path of the file under transfer.
- GroupID – The T/F bit suggests whether the GroupID is valid or not. If valid, then this can be any number from 1 to 128 (7-bit). GroupID 0 is the default group.
- Update Timeout - On Communication failure for a period greater than this timeout value, the Client MUST exit from the current process and become available for next update cycle without participating for the rest of the current update process. The status of the client will remain 'FAIL' until the beginning of the next update cycle. Refer to Appendix A for default UPDATE_TIMEOUT value.
- Variable Length Data – String of variable length, the length is specified by the previous field. For example, File Name Length field specifies the length of this field.

a. Using the GroupID field, the interested clients having the same GroupID need to join the multicast group as specified in the notification message packet. Please refer to Table 6 for the message format.

b. Server can force the clients to upgrade their firmware by sending the sub code FORCE_UPGRADE. Upon receiving this message from the server, the clients compulsorily need to participate in the MSU process.

c. The file [FA] is divided into smaller units called Chunks.

d. The chunk is divided into smaller units called SN.

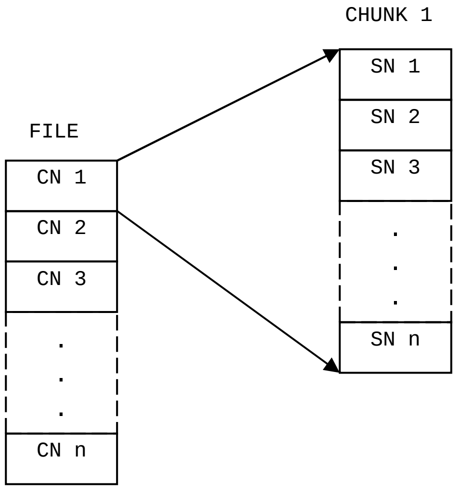


Figure 2 File – Chunk - Sequence Structure

e. Server prepares the chunks of the file which it needs to send over the network to the clients. Refer Figure 2.

f. The chunks are further divided into a number of parts depending on the maximum datagram payload size, optimization requirements and datagram fragmentation. These numbers are the sequence numbers [SN]. Refer Figure 2. Please refer to Number of Chunks, Sequence Number Limit and Size of Sequence fields of Table 6.

g. Server waits for a predefined JOIN_MSU_WAIT_TIME time (refer Appendix A) for the clients to join the multicast group.

h. Server starts sending the data packet from the first chunk to the end of the chunk.

i. While sending each chunk of data, the server inserts the SN in the message. The data transfer message format is shown in the Table 7.

<table>
<tr><th>Byte-0</th><th>Byte-1</th><th>Byte-2</th><th>Byte-3</th></tr>
<tr><td>Opcode (4) | Subcode (4)</td><td colspan="2">File Number</td><td>IP Version (1) | Protocol Version (3) | reserved (4)</td></tr>
<tr><td colspan="4">Chunk Number [CN]</td></tr>
<tr><td>Sequence Number (Min 1 - Max 32)</td><td colspan="2">Data Length</td><td>File/Chunk Transfer States (2)| reserved (6)</td></tr>
<tr><td colspan="4">Variable Length Data</td></tr>
</table>

Table 7 - Data Transfer Message Format (GM)

- Opcode – MSU_DEVICE_UPDATE (1)
- Subcode – DATA_TRANSFER (4)

Proprietary Notice: This document contains proprietary information of Schneider Electric and neither the document nor said proprietary information shall be published, reproduced, copied, disclosed or used for any purpose other than consideration of this document without the express written permission of a duly authorized representative of said company. Patent Application No. 1448/CHE/2011 – 26 April 2011

- File Number - Refer Table 6 description.
- IP Version – Refer Table 1 description.
- Protocol Version- Refer Table 1 description.
- CN – Refers to the chunk number that is being transferred. It starts from 1.
- SN – Refers to the sequence number that is sent in this message. It starts from 1, maximum being Sequence Number Limit. Please refer Table 6 for Sequence Number Limit.
- Data Length – Length of the variable length data in bytes.
- File/Chunk Transfer States – FILE_START [0]/END[1] – BIT 0
  CHUNK_START [0]/END [1] – BIT 1
- Variable Length Data – The actual FA data that needs to be transferred.

j. The client builds a database of all the frames received with CN and SN information.

k. After sending one complete chunk, the Server waits for a predefined SCM_WAIT_TIME time (refer Appendix A) for the SCM. In this mode, the clients request the server all the frames that were missed. These missed frames are indicated by the SN. The server re-transmits all the missed frames on the Multicast Address (optionally, if CM Multicast Address is supported, then, this address will be used). The Unicast SCM Message Format is shown in Table 8.

<table>
<tr><th>Byte-0</th><th>Byte-1</th><th>Byte-2</th><th>Byte-3</th></tr>
<tr><td>Opcode (4) | Subcode (4)</td><td colspan="2">File Number</td><td>IP Version (1) | Protocol Version (3) | reserved (4)</td></tr>
<tr><td colspan="4">Chunk Number [CN]</td></tr>
<tr><td>Number of Sequences</td><td colspan="3">Reserved</td></tr>
<tr><td colspan="4">Bit Map of Sequences</td></tr>
</table>

Table 8 - SCM Message Format (IM)

- Opcode – MSU_DEVICE_UPDATE (1)
- Subcode – SEQUENCE_COMPLAIN_MODE (5)
- File Number - Refers to the number of the file to which this SCM belongs.
- IP Version – Refer Table 1 for description.
- Protocol Version- Refer Table 1 for description.
- CN – Refers to the number of the chunk to which this SCM belongs.
- Number of Sequences – The total number of missed frames.
- Bit Map of Sequences – The missed sequence numbers are represented by bits. The 0th bit indicates SN=1 and so on.

l. After the SCM_WAIT_TIME has elapsed, the server sends the SCM completed message for the current chunk which it has just transferred. The message format is shown in Table 9. If the

SCM_Retry_Flag field is non-zero, then the client has another time interval of SCM_WAIT_TIME to form and send SCM message for those sequences it still has not received. This process of SCM retries continue until the server sends a SCM completed message with SCM_Retry_Flag field set to zero after which any further SCM messages from the client will be ignored by the server for the current chunk.

<table>
<tr><th>Byte-0</th><th>Byte-1</th><th>Byte-2</th><th>Byte-3</th></tr>
<tr><td>Opcode (4) | Subcode (4)</td><td colspan="2">File Number</td><td>IP Version (1) | Protocol Version (3) | reserved (4)</td></tr>
<tr><td>SCM_Retry_Flag</td><td colspan="3">Reserved</td></tr>
</table>

Table 9 - SCM Completed Message Format (GM)

- Opcode – MSU_DEVICE_UPDATE (1)
- Subcode – SCM_TRANSFER_COMPLETED (10)
- File Number – Refers to the index of the file that was transferred.
- IP Version – Refer Table 1 description.
- Protocol Version- Refer Table 1 for description.

m. Server re-transmits all the SCM requested packets. The clients receive this re-transmitted data and recover the malformed chunk. If there are no packets requested by any of the clients, the server proceeds with the transfer of the next chunk.

n. Server repeats the same procedure until all the chunks are sent, including the last chunk.

o. After the SCM process for the last chunk, the server must send the Transfer Completed Message with the Subcode set to TRANSFER_COMPLETED. Setting File/Chunk Transfer States field to FILE_COMPLETED is OPTIONAL. The Multicast Transfer Completed Message Format is shown in Table 10.

p. The number of times the server sends the Transfer Completed message is implementation specific.

<table>
<tr><th>Byte-0</th><th>Byte-1</th><th>Byte-2</th><th>Byte-3</th></tr>
<tr><td>Opcode (4) | Subcode (4)</td><td colspan="2">File Number</td><td>IP Version (1) | Protocol Version (3) | reserved (4)</td></tr>
</table>

Table 10 - Transfer Completed Message Format (GM)

- Opcode – MSU_DEVICE_UPDATE (1)
- Subcode – TRANSFER_COMPLETED (7)
- File Number – Refers to the index of the file that was transferred.
- IP Version – Refer Table 1 description.
- Protocol Version- Refer Table 1 for description.

q. After the complete file transfer including the SCM for the last chunk and after the server sends the Transfer Completed Message, the MSU process goes into CCM. In this mode, the clients request for complete chunk re-transmission by sending the missed chunk number using the CCM Message. The server gathers the information from the clients within a window of time meant for CCM. The Unicast CCM Message Format is shown in Table 11.

Byte-0	Byte-1	Byte-2	Byte-3
Opcode (4) | Subcode (4)	File Number		IP Version (1) | Protocol Version (3) | reserved (4)
Number of Chunks			
Chunk Number [CN]			
…			
Variable Length CNs			

Table 11 - CCM Message Format (IM)

- Opcode - MSU_DEVICE_UPDATE (1)
- Subcode – CHUNK_COMPLAIN_MODE (6)
- File Number - Refers to the number of the file to which this CCM belongs.
- IP Version – Refer Table 1 description.
- Protocol Version - Refer Table 1 description.
- Number of Chunks – This is the number of missed/malformed chunks.
- CN – Refers to the missed chunk number that the client wants the server to re-transmit. Each chunk number takes four bytes in the message format stacked one below the other as indicated by Variable Length CNs.
- Variable Length CNs – Refer to CN field.

r. Server waits for a predefined CCM_WAIT_TIME time (refer Appendix A) allocated for the CCM. After this time, the server re-transmits the requested chunks one by one in ascending order until all the chunks are transmitted. The frame format and the procedure for sending the chunks remains same as before including the SCM states.

s. After receiving Transfer completed message and the CCM, the clients verify the sanity of the received file using CRC. Optionally, the clients can verify CRC immediately after receiving the last chunk of the file if they do not need to participate in the CCM. This step is implementation specific.

t. The server sends the CCM completed message after every iteration of the CCM. The number of iterations of the CCM is implementation specific. This number is predefined and known to the server and the clients. The format of this message is shown in Table 12. The CCM cycle is carried out more than once to increase the success rate of file transfer.

<table>
<tr><th>Byte-0</th><th>Byte-1</th><th>Byte-2</th><th>Byte-3</th></tr>
<tr><td>Opcode (4) |
Subcode (4)</td><td colspan="2">File Number</td><td>IP Version (1) | Protocol Version (3) |
reserved (4)</td></tr>
</table>

Table 12 - CCM Completed Message Format (GM)

- Opcode – MSU_DEVICE_UPDATE (1)
- Subcode – CCM_TRANSFER_COMPLETED (9)
- File Number – Refers to the index of the file that was transferred.
- IP Version – Refer Table 1 description.
- Protocol Version- Refer Table 1 for description.

u. At any point of time during the MSU process, the Server can send MSU abort message to all the participating clients to exit from the current MSU process by sending the TRANSFER_ABORTED message as shown in Table 13.

<table>
<tr><th>Byte-0</th><th>Byte-1</th><th>Byte-2</th><th>Byte-3</th></tr>
<tr><td>Opcode (4) |
Subcode (4)</td><td colspan="2">File Number</td><td>IP Version (1) | Protocol Version (3) |
reserved (4)</td></tr>
</table>

Table 13 - Transfer Aborted Message Format (GM)

- Opcode – MSU_DEVICE_UPDATE (1)
- Subcode – TRANSFER_ABORTED (8)
- File Number – Refers to the index of the file that was transferred.
- IP Version – Refer Table 1 description.
- Protocol Version- Refer Table 1 for description.

v. If the clients receive MSU abort, they have to undo all the MSU process which is subject to a clients' local matter.

w. Status Message – This message is sent (unicast) by the clients to the server with the Subcode set to CLIENT_STATUS_UPDATE_RESPONSE. Two packets (number of packets is implementation specific) are sent in succession with a delay of STATUS_MESSAGE_GAP after the completion of CRC calculation.

The server can explicitly request for this message from the clients by sending (unicast) with the Subcode CLIENT_STATUS_UPDATE_REQUEST. Refer Table 14 for the message f

<table>
<tr><th>Byte-0</th><th>Byte-1</th><th>Byte-2</th><th>Byte-3</th></tr>
<tr><td>Opcode (4) |
Subcode (4)</td><td>Status</td><td>Error Code</td><td>IP Version (1) | Protocol Version (3) | reserved (4)</td></tr>
<tr><td colspan="4">TransactionID</td></tr>
<tr><td colspan="4">DeviceID</td></tr>
<tr><td>CCM Retry</td><td colspan="3">Reserved</td></tr>
</table>

Table 14 – Status Message Format (IM)

Proprietary Notice: This document contains proprietary information of Schneider Electric and neither the document nor said proprietary information shall be published, reproduced, copied, disclosed or used for any purpose other than consideration of this document without the express written permission of a duly authorized representative of said company. Patent Application No. 1448/CHE/2011 – 26 April 2011

- Opcode - MSU_DEVICE_UPGRADE (1)
- Subcode – CLIENT_STATUS_UPDATE_REQUEST (11)
  CLIENT_STATUS_UPDATE_RESPONSE (12)
- Status - PASS (0)
  FAIL (1)
  IN_PROGRESS (3)

- Error Code - Failure code (0 if there is no failure)
- IP Version – Refer Table 1 description.
- Protocol Version - Refer Table 1 description.
- TransactionID – The latest TransactionID stored by the client for the current MSU cycle.
- Device ID – Device Identification
- CCM Retry – Number of times the client had participated in CCM processes.

13. Note

a. Upon the completion of each chunk, only those clients go into the SCM that has missed any packet in that chunk sent by the server.

b. Upon the completion of all the chunks, only those clients go into the CCM that has missed any chunk sent by the server.

c. Clients that have received complete chunk must ignore the SCM/CCM transactions.

d. Server collects the packet numbers from the clients during the SCM and CCM and form a sequence of all the missed packets/chunks. Forming of sequence is done to avoid the overlapping of missed packets/chunks of multiple clients. This avoids the multiple re-transmission of the same packet/chunk to more than one client.

e. During CCM, the server sends the requested malformed chunks to clients, one chunk at a time until all the requested chunks get transferred.

f. The rules of SCM and CCM apply as it is for all the chunks irrespective of whether the chunk is in the normal MSU process or in the malformed chunk re-transmission process.

g. The clients which are participating in the malformed chunk re-transmission process have to analyze by the end of each re-transmitted chunk if they have any more malformed chunks still left.

h. Clients must discard duplicate packets received on the network.

i. Clients must handle out of sequence reception of packets based on the SN.

j. This protocol is not restricted to firmware. It may be used for any file on the client. That is, the configuration file, file system, special files etc.

14. Limitations of the MSU Protocol

a. Supports only IPV4
b. Server and Client devices must support file system

Appendix A

NAME	DEFAULT VALUES*	UNITS
JOIN_MSU_WAIT_TIME	5	seconds
SCM_WAIT_TIME	3	seconds
CCM_WAIT_TIME	5	Seconds
STATUS_MESSAGE_GAP	10	Milliseconds
UPDATE_TIMEOUT	10	seconds

*The values indicated are configurable and are a local matter of the products that are participating in the MSU process.

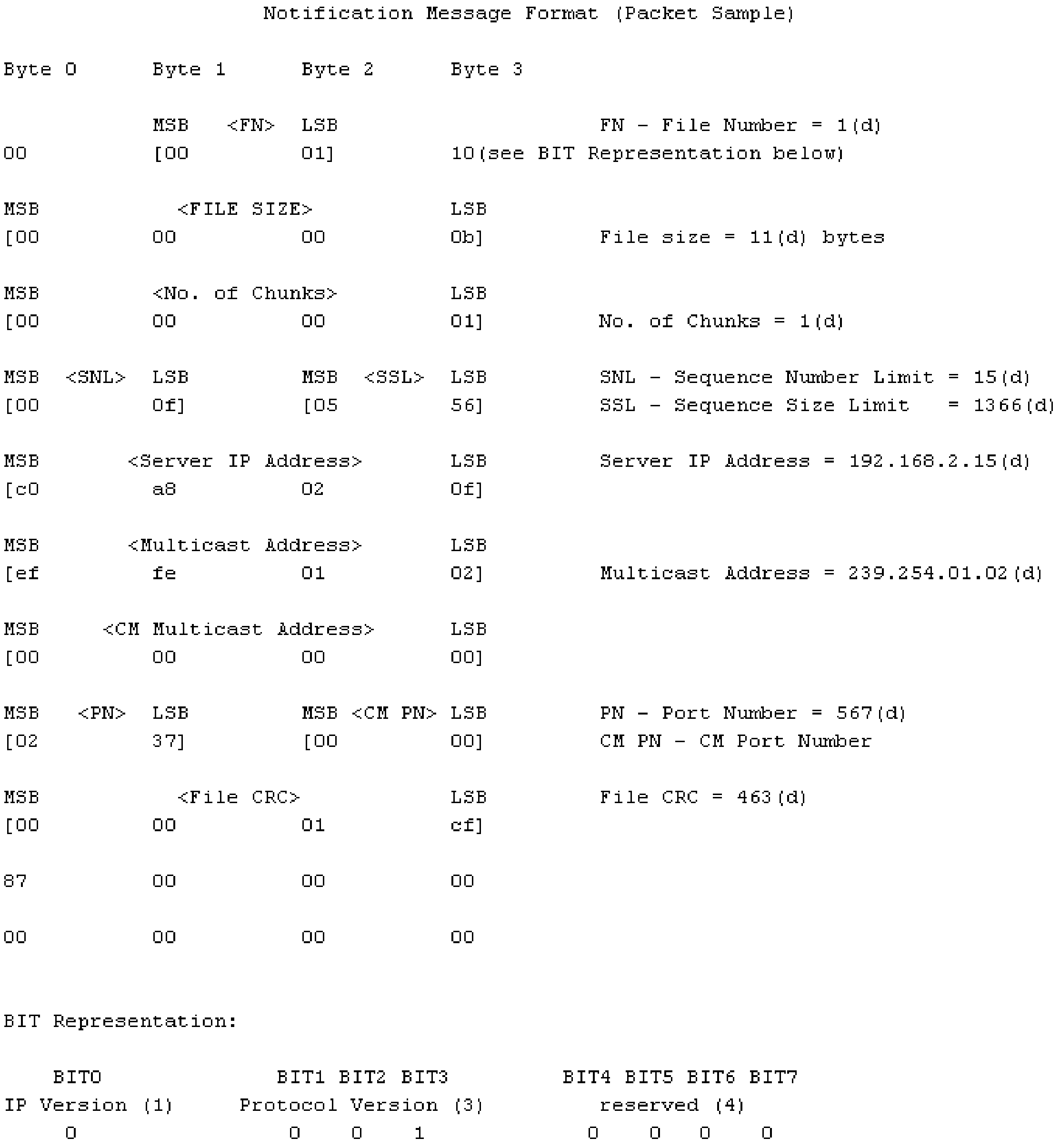
Table 15 – Default Values

No.	OpCode	SubCode
1	MSU_DEVICE_CODE_UNDEF (0)	NA
2	MSU_DEVICE_UPGRADE (1)	UPGRADE (1)
		DOWNGRADE (2)
		FORCE_UPGRADE (3)
		DATA_TRANSFER (4)
		SEQUENCE_COMPLAIN_MODE (5)
		CHUNK_COMPLAIN_MODE (6)
		TRANSFER_COMPLETED (7)
		TRANSFER_ABORTED (8)
		CCM_TRANSFER_COMPLETED (9)
		SCM_TRANSFER_COMPLETED (10)
		CLIENT_STATUS_UPDATE_REQUEST (11)
		CLIENT_STATUS_UPDATE_RESPONSE (12)
3	MSU_DEVICE_DISCOVER (2)	WHO-IS (1)
		I-AM (2)
4	MSU_DEVICE_GROUP (3)	SET_GROUP_ID (1)
		RESET_GROUP_ID (2)
		CREATE_GROUP_ACK (3)
5	MSU_USER_CMD (4)	START (1)
		STOP (2)
		INIT (3)

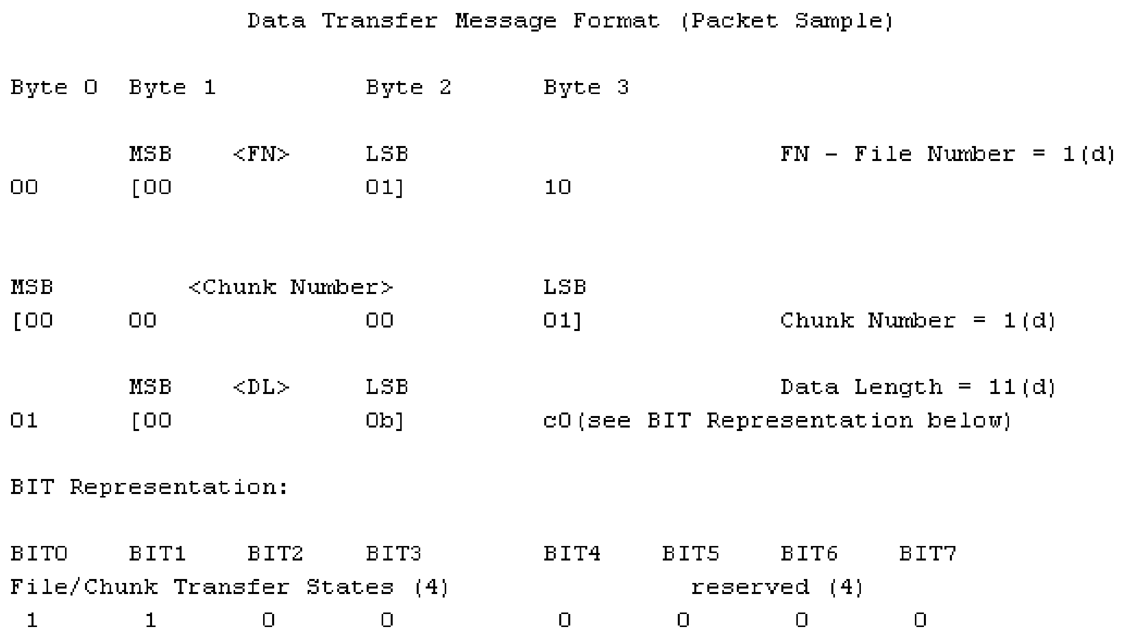
Table 16 – Command List

Appendix B

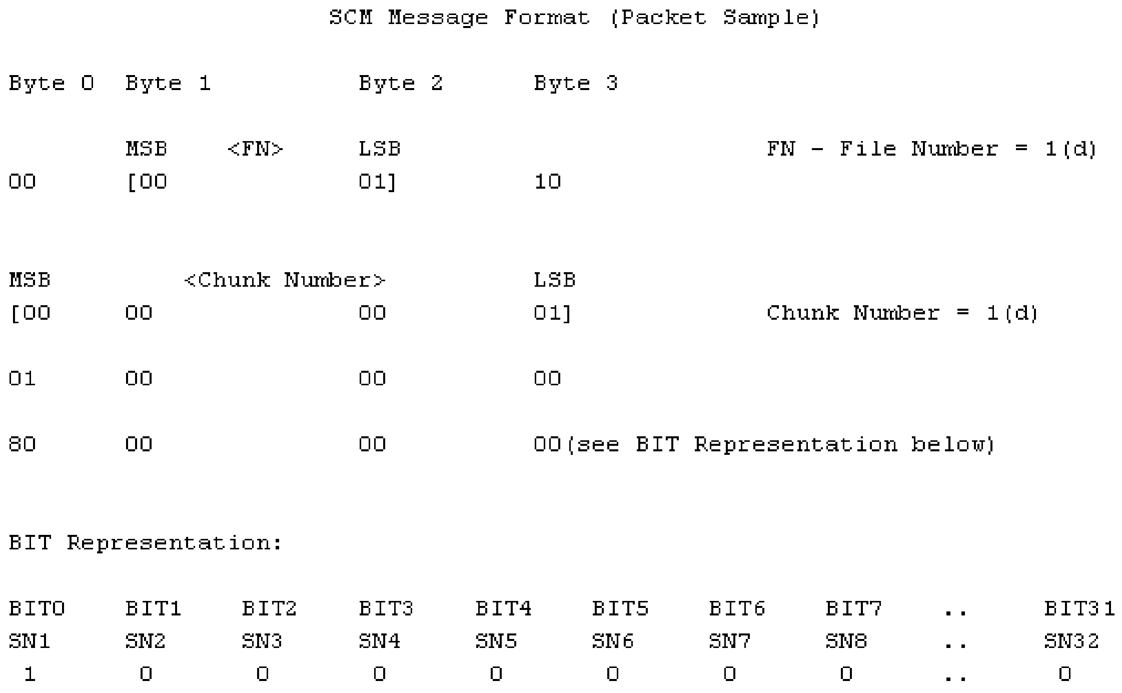
Appendix B represents samples of Notification Message Format as per Table 6**; Data Transfer Message Format as per** Table 7**; SCM Message Format as per** Table 8**; CCM Message Format as per** Table 11**. These formats are illustrated as seen on the wire.**



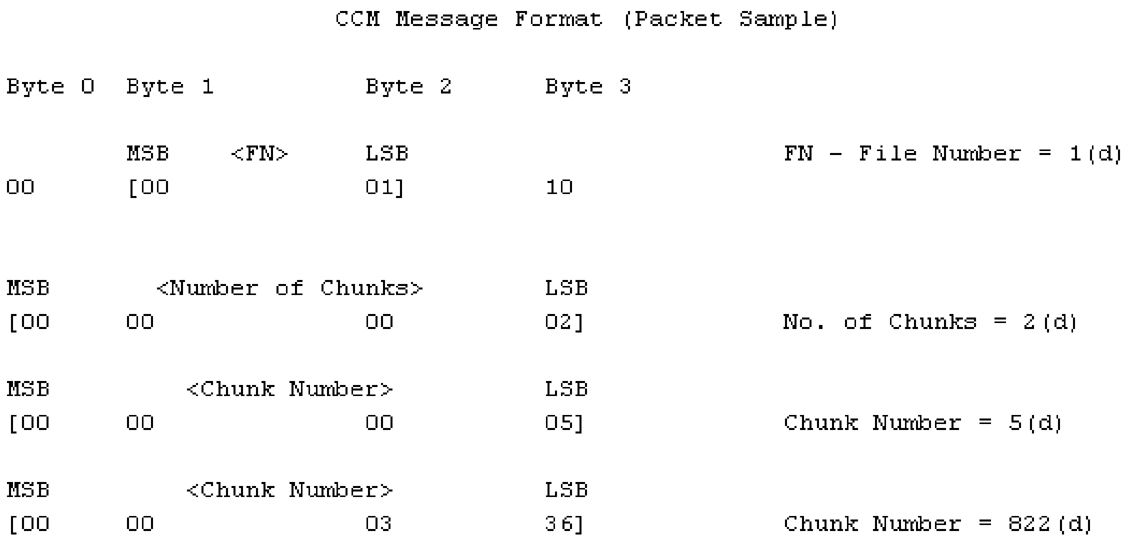
```
                    Notification Message Format (Packet Sample)

Byte 0      Byte 1      Byte 2      Byte 3

            MSB   <FN>  LSB                     FN - File Number = 1(d)
00          [00         01]         10(see BIT Representation below)

MSB           <FILE SIZE>           LSB
[00         00          00          0b]         File size = 11(d) bytes

MSB         <No. of Chunks>         LSB
[00         00          00          01]         No. of Chunks = 1(d)

MSB  <SNL>  LSB         MSB  <SSL>  LSB         SNL - Sequence Number Limit = 15(d)
[00         0f]         [05         56]         SSL - Sequence Size Limit   = 1366(d)

MSB       <Server IP Address>       LSB         Server IP Address = 192.168.2.15(d)
[c0         a8          02          0f]

MSB       <Multicast Address>       LSB
[ef         fe          01          02]         Multicast Address = 239.254.01.02(d)

MSB     <CM Multicast Address>      LSB
[00         00          00          00]

MSB   <PN>  LSB         MSB <CM PN> LSB         PN - Port Number = 567(d)
[02         37]         [00         00]         CM PN - CM Port Number

MSB           <File CRC>            LSB         File CRC = 463(d)
[00         00          01          cf]

87          00          00          00

00          00          00          00


BIT Representation:

    BIT0             BIT1 BIT2 BIT3         BIT4 BIT5 BIT6 BIT7
IP Version (1)     Protocol Version (3)        reserved (4)
     0                 0    0    1            0    0   0    0
```



```
                   Data Transfer Message Format (Packet Sample)

Byte 0  Byte 1          Byte 2      Byte 3

        MSB    <FN>     LSB                         FN - File Number = 1(d)
00      [00             01]         10


MSB         <Chunk Number>          LSB
[00     00              00          01]             Chunk Number = 1(d)

        MSB    <DL>     LSB                         Data Length = 11(d)
01      [00             0b]         c0(see BIT Representation below)

BIT Representation:

BIT0    BIT1    BIT2    BIT3        BIT4    BIT5    BIT6    BIT7
File/Chunk Transfer States (4)              reserved (4)
 1       1       0       0           0       0       0       0
```



```
                        SCM Message Format (Packet Sample)

Byte 0  Byte 1          Byte 2      Byte 3

        MSB    <FN>     LSB                         FN - File Number = 1(d)
00      [00             01]         10


MSB         <Chunk Number>          LSB
[00     00              00          01]             Chunk Number = 1(d)

01      00              00          00

80      00              00          00(see BIT Representation below)


BIT Representation:

BIT0    BIT1    BIT2    BIT3    BIT4    BIT5    BIT6    BIT7    ..     BIT31
SN1     SN2     SN3     SN4     SN5     SN6     SN7     SN8     ..     SN32
 1       0       0       0       0       0       0       0      ..      0
```



```
                       CCM Message Format (Packet Sample)

Byte 0  Byte 1          Byte 2      Byte 3

        MSB    <FN>     LSB                         FN - File Number = 1(d)
00      [00             01]         10


MSB       <Number of Chunks>        LSB
[00     00              00          02]             No. of Chunks = 2(d)

MSB         <Chunk Number>          LSB
[00     00              00          05]             Chunk Number = 5(d)

MSB         <Chunk Number>          LSB
[00     00              03          36]             Chunk Number = 822(d)
```

Proprietary Notice: This document contains proprietary information of Schneider Electric and neither the document nor said proprietary information shall be published, reproduced, copied, disclosed or used for any purpose other than consideration of this document without the express written permission of a duly authorized representative of said company. Patent Application No. 1448/CHE/2011 – 26 April 2011

Appendix C

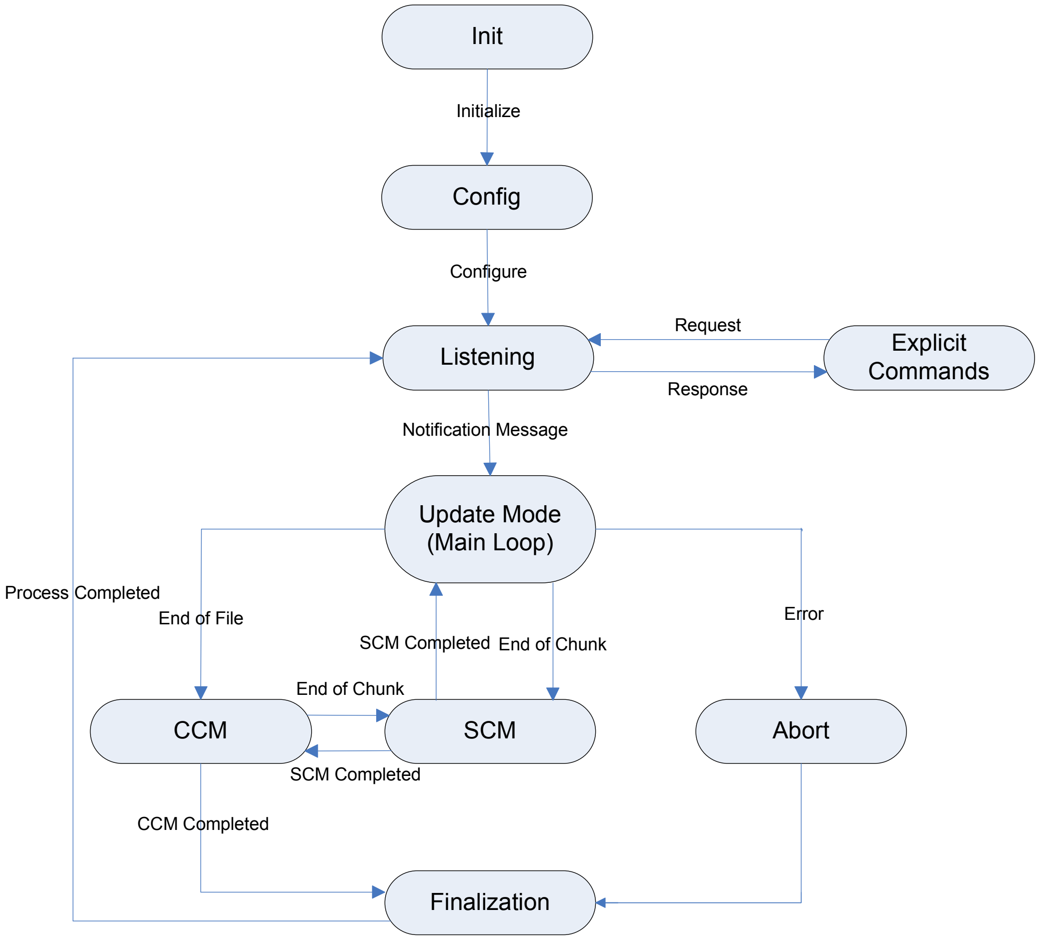


Figure 3 - Client Flow Chart

Init	– Initialization operation
Config	– Configuration operation
Listening	– Mode of listening to Notification message
Update Mode	– Main mode of MSU operation
Explicit Commands	– Who-Is, I-am, Status commands, etc.
CCM	- Chunk Complain Mode of operation
SCM	- Sequence Complain Mode of operation
Abort	- Abort operation handling
Finalization	- CRC calculation, status determination