TECHNICAL NOTE

MD8470A
Signalling Tester

Application Test

ANRITSU CORPORATION
Detailed Explanations for Each Application

Contents

Part 1: Packet Communication (HTML and WAP)
Part 2: Video Streaming
Part 3: Video Telephony
Part 4: SMS
Part 5: MMS
Part 6: DRM
Part 1: Packet Communication (HTTP and WAP)

Packet Communication (HTTP)

Protocol Stack

HTTP : Hypertext Transfer Protocol
TCP : Transmission Control Protocol
UDP : User Datagram Protocol
IP : Internet Protocol

L2 (W-CDMA) : MAC / RLC
  MAC : Media Access Control
  RLC : Radio Link Control

L2 (GSM/GPRS) : DL /
  DL : Data Link

L1 (W / G) : PHY(Physical Layer)
Packet Communication (HTTP)

- **DNS (Domain Name System)**
  - DNS supports the conversion between a domain name and an IP address.
  - (Ex.: vodafone.com <--> 192.168.1.2)

- **IIS (Internet Information Server)**
  - IIS supports the Web server function that sends the contents and the default contents (default web site).

---

**Protocol Sequence**

**MS**

- HTTP
- TCP / UDP
- IP

**NW**

- PDP Activation (Wireless Protocol)
- ARP Request(IP)
- ARP Reply(IP)
- Connection Request(TCP)
- Connection Ack & Request(TCP)
- Connection Ack(TCP)
- Get(HTTP)
- OK(Response: HTTP)
- Ack(TCP)
- Disconnect(TCP)
- PDP De-activation (Wireless Protocol)
### Packet Communication (HTTP)

**Protocol Sequence (ARP Request)**

1. **HTTP**
2. **TCP / UDP**
3. **IP**
4. **L3** (Wireless Protocol)
5. **L2** (Wireless Protocol)
6. **L1** (Wireless Protocol)

1. MS (IP : 192.168.1.1) searches the MAC address of the Server (IP : 192.168.1.2)

**Protocol Sequence (ARP Reply)**

1. **HTTP**
2. **TCP / UDP**
3. **IP**
4. **L3** (Wireless Protocol)
5. **L2** (Wireless Protocol)
6. **L1** (Wireless Protocol)

1. Server (IP : 192.168.1.2) replies the MAC address to MS (IP : 192.168.1.1)
Packet Communication (HTTP)

Protocol Sequence (Connection Req.)

- **L1 (Wireless Protocol)**
- **L2 (Wireless Protocol)**
- **IP**
- **TCP / UDP**
- **HTTP**

*1: data from ‘6616(port number of MS)’ to ‘80(http port number of the Server)’

*2: connection request from MS to server

---

Packet Communication (HTTP)

Protocol Sequence (Conn.Ack & Req.)

- **L1 (Wireless Protocol)**
- **L2 (Wireless Protocol)**
- **IP**
- **TCP / UDP**
- **HTTP**

*1: data from ‘80(http port number of the Server)’ to ‘6616(port number of MS)’

*2: Ack of ‘connection request’ and connection request from server to MS.
Packet Communication (HTTP)

Protocol Sequence (Connection Ack)

- HTTP
- TCP / UDP
- IP
- L3
- L2 (Wireless Protocol)
- L1 (Wireless Protocol)

*1: data from '6616(port number of MS)' to '80(http port number of the Server)'

*2: Ack of 'connection request'

Packet Communication (HTTP)

Protocol Sequence (HTTP Get)

- HTTP
- TCP / UDP
- IP
- L3
- L2 (Wireless Protocol)
- L1 (Wireless Protocol)

*1: request the html contents. '/' means the relative path. Only '/' case is root contents of the server.

*2:
- Host: Server IP address
- Application: supported MIME type
- Language: en (English)
- Charaset: utf-8 (one of the common unicode)
Packet Communication (HTTP)

Protocol Sequence (HTTP Response)

- HTTP
- TCP / UDP
- IP
- L3
- L2 (Wireless Protocol)
- L1 (Wireless Protocol)

*1: 'Http/1.1 200 OK' means 'success of downloading'
- 'Content-Location' : path of the content
- 'Content-Type' : MIME type of the content
- 'Content-Length' : Length of the content

*2: HTML content data itself

---

Packet Communication (HTTP)

Protocol Sequence (TCP Ack)

- HTTP
- TCP / UDP
- IP
- L3
- L2 (Wireless Protocol)
- L1 (Wireless Protocol)

*1: TCP level acknowledge of previous HTTP content access
Packet Communication (HTTP)

Protocol Sequence (TCP Disconnect)

- HTTP
- TCP / UDP
- IP
- L3
- L2 (Wireless Protocol)
- L1 (Wireless Protocol)

*1 : Disconnect request

Packet Communication (WAP)

Protocol Stack

- WSP : Wireless Session Protocol
- WTP : Wireless Transaction Protocol
- WDP : Wireless Datagram Protocol
- IP : Internet Protocol

- L2 (W-CDMA) : MAC / RLC
  MAC : Media Access Control
  RLC : Radio Link Control

- L2 (GSM/GPRS) : DL /
  DL : Data Link

- L1 (W / G) : PHY (Physical Layer)
Packet Communication (WAP)

- **WAP Gateway**
  - Converting to the format supported by the UE and downsizing the contents
  - MD8470A uses ‘Kannel’ free software V1.4

All these parts are available in an external PC

- File path (default contents) can be defined in the IIS

Packet Communication (WAP-HTML)

**Protocol Sequence**

- MS
  - WSP
  - WTP
  - WDP
  - IP

- L3

- L2 (Wireless Protocol)

- L1 (Wireless Protocol)

- NW
  - PDP Activation (Wireless Protocol)
  - ARP Request (IP)
  - ARP Reply (IP)
  - WSP Connect (WSP)
  - WSP Connect Reply (WSP)
  - WTP Ack & WSP Get (WTP)
  - WTP Result (WTP)
  - WTP Ack (WTP)
  - WSP Disconnect (WSP)
  - PDP De-activation (Wireless Protocol)
Packet Communication (WAP-HTML)

Protocol Sequence (ARP Request)

*1 : MS(IP : 192.168.1.1) searches the MAC address of the Server(IP : 192.168.1.2)

Protocol Sequence (ARP Reply)

*1 : Server (IP : 192.168.1.2) replies the MAC address to MS(IP : 192.168.1.1)
Packet Communication (WAP-HTML)

Protocol Sequence (WSP Connect)

1. WTP ‘PDU Type: Invoke’ : request to WSP (Transaction ID = 1)
2. WSP ‘PDU Type: Connect’ : WSP connect WSP version is 1.0
3. Size of sending in 1 time
   - Server SDU Size: 356352
   - Client SDU Size: 344250

Protocol Sequence (WSP Conn. Reply)

1. WTP ‘PDU Type: Result’ : reply from WSP (Transaction ID = 1)
2. WSP ‘PDU Type: ConnectReply’ : WSP Connect Reply
Packet Communication (WAP-HTML)

**Protocol Sequence (WTP Ack & WSP Get)**

1. **First WTP ‘PDU Type : Ack’**: Ack of previous Reply (Transaction ID = 1)
2. **Second WTP ‘PDU Type : Invoke’**: request to WSP (request of ‘WSP Get’, Transaction ID = 2)

---

*1* : WSP ‘PDU Type : Get’ : request the content

*2* : WSP > Headers >
- Accept : supported MIME type
- Accept-Language : English
- Accept-Charset : utf-8 (common unicode)
Packet Communication (WAP-HTML)

**Protocol Sequence (WTP Result)**

1. **WTP 'PDU Type : Result'**: reply from WSP (Transaction ID = 2)
2. **'Trailer Flags : Not Last Packet'**: means the segmentation of content data had happened. Downloading content is still continuing.

**Protocol Sequence (WSP Reply)**

1. **WTP 'PDU Type : Segmented Result'**: reply from WSP (Transaction ID = 2) for the second data block
2. **'Trailer Flags : Last Packet of message'**: means the last part of the content data.
Packet Communication (WAP-HTML)

**Protocol Sequence (WSP Reply)**

1. **WSP 'PDU Type : Reply'**: reply the content access
   - 'Status : 200 OK' means success of download

2. **content location and length**

3. **html data itself**

**Protocol Sequence (WTP Ack)**

1. **WSP 'PDU Type : Ack'**: Ack for the WTP result of first content data block (Transaction ID = 2)
Packet Communication (WAP-HTML)

Protocol Sequence (WTP Ack)

- **WSP**
- **WTP**
- **WDP**
- **IP**
- **L3**
- **L2 (Wireless Protocol)**
- **L1 (Wireless Protocol)**

*1 : WSP ‘PDU Type : Ack’ : Ack for the WTP result of second(last) content data block (Transaction ID = 2)

---

Packet Communication (WAP-HTML)

Protocol Sequence (WSP Disconnect)

- **WSP**
- **WTP**
- **WDP**
- **IP**
- **L3**
- **L2 (Wireless Protocol)**
- **L1 (Wireless Protocol)**

*1 : WTP ‘PDU Type : Invoke’ : request to WSP (request of ‘WSP Disconnect’, Transaction ID = 3)

*2 : WSP ‘PDU Type : Disconnect’ : disconnection
Packet Communication (WAP-WML)

**Protocol Sequence**

MS

PDP Activation (Wireless Protocol)

ARP Request(IP)

ARP Reply(IP)

WSP Connect(WSP)

WSP Connect Reply(WSP)

WTP Ack & WSP Get(WTP)

WTP Result(WTP)

WSP Reply(WSP)

WTP Ack(WTP)

WTP Ack & WSP Get(WTP)

WSP Reply(WSP)

WTP Ack(WTP)

WSP Disconnect(WSP)

PDP De-activation (Wireless Protocol)

NW

**Protocol Sequence (ARP Request)**

*1: MS(IP : 192.168.1.1) searches the MAC address of the Server(IP : 192.168.1.2)**
Packet Communication (WAP-WML)

**Protocol Sequence (ARP Reply)**

*1 : Server (IP : 192.168.1.2) replies the MAC address to MS (IP : 192.168.1.1)

**Protocol Sequence (WSP Connect)**

*1 : WTP ‘PDU Type : Invoke’ : request to WSP (Transaction ID = 5)

*2 : WSP ‘PDU Type : Connect’ : WSP connect WSP version is 1.0

*3 : Size of sending in 1 time

Server SDU Size : 356352
Client SDU Size : 344250
Packet Communication (WAP-WML)

**Protocol Sequence (WSP Conn. Reply)**

1. **WTP** 'PDU Type : Result' : reply from WSP (Transaction ID = 5)
2. **WSP** 'PDU Type : ConnectReply' : WSP Connect Reply

**Protocol Sequence (WTP Ack & WSP Get)**

1. First WTP 'PDU Type : Ack' : Ack of previous Reply (Transaction ID = 5)
2. Second WTP 'PDU Type : Invoke' : request to WSP (request of 'WSP Get', Transaction ID = 6)
Packet Communication (WAP-WML)

Protocol Sequence (WTP Ack & WSP Get)

1. WSP ‘PDU Type : Get’ : request the content
   URI : path of content
2. WSP > Headers >
   Accept : supported MIME type
   Accept-Language : English
   Accept-Charset : utf-8 (common unicode)

L3
L2 (Wireless Protocol)
L1 (Wireless Protocol)

Packet Communication (WAP-WML)

Protocol Sequence (WTP Result)

1. WTP ‘PDU Type : Result’ : reply from WSP (Transaction ID = 6)
2. ‘Trailer Flags : Not Last Packet’ means the segmentation of content data had happened
   Downloading content is still continuing

L3
L2 (Wireless Protocol)
L1 (Wireless Protocol)
Packet Communication (WAP-WML)

**Protocol Sequence (WSP Reply)**

1. **WTP 'PDU Type : Segmented Result'**: reply from WSP (Transaction ID = 6) for the second data block.
2. **'Trailer Flags : Last Packet of message'** means the last part of the content data.

**Packet Communication (WAP-WML)**

**Protocol Sequence (WSP Reply)**

1. **WSP 'PDU Type : Reply'**: reply the content access.
   - "Status : 200 OK" means success of download.
2. **Content length**
3. **WML data itself**
Packet Communication (WAP-WML)

Protocol Sequence (WTP Ack)

1. WSP 'PDU Type : Ack' : Ack for the WTP result of first content data block (Transaction ID = 6)

Protocol Sequence (WTP Ack & WSP Get)

1. WSP 'PDU Type : Ack' : Ack for the WTP result of second(last) content data block (Transaction ID = 6)
2. Second WTP 'PDU Type : Invoke' : request to WSP (request of ‘WSP Get’, Transaction ID = 7)
Packet Communication (WAP-WML)

Protocol Sequence
(WTP Ack & WSP Get)

1. **WSP ‘PDU Type : Get’**: request the content
   
   * URI: path of content (md8470a.wbmp)
   * UE uses ‘Get’ for each contents file (this logic is same as HTTP)

2. **WSP > Headers**:
   - Accept: supported MIME type
   - Accept-Language: English
   - Accept-Charset: utf-8 (common unicode)

**Notes:**
- *1: WSP ‘PDU Type : Get’ : request the content
- *2: URI: path of content (md8470a.wbmp)
- *3: UE uses ‘Get’ for each contents file (this logic is same as HTTP)

---

Packet Communication (WAP-WML)

Protocol Sequence
(WSP Reply)

1. **WTP ‘PDU Type : Result’**: reply from WSP
   - (Transaction ID = 7)

2. **WSP ‘PDU Type : Reply’**: reply the content access
   - ‘Status : 200 OK’ means success of download

3. **Content-length**: the length of the content

**Notes:**
- *1: WTP ‘PDU Type : Result’ : reply from WSP
- *2: WSP ‘PDU Type : Reply’ : reply the content access
- *3: ‘Status : 200 OK’ means success of download
Packet Communication (WAP-WML)

Protocol Sequence (WTP Ack)

1. WSP ‘PDU Type : Ack’ : Ack for the WTP result of the content data (Picture data) (Transaction ID = 7)

Protocol Sequence (WSP Disconnect)

1. WTP ‘PDU Type : Invoke’ : request to WSP (request of ‘WSP Disconnect’), Transaction ID = 3
2. WSP ‘PDU Type : Disconnect’ : disconnection
Part 2: Video Streaming

Video Streaming (RTSP)

Protocol Stack

- SDP: Session Description Protocol
- RTP: Real-time Transfer Protocol
- RTCP: Real-time Transfer Control Protocol
- RTSP: Real Time Streaming Protocol
- UDP: User Datagram Protocol
- TCP: Transmission Control Protocol
- IP: Internet Protocol
- L2(W-CDMA): MAC / RLC
  - MAC: Media Access Control
  - RLC: Radio Link Control
- L2(GSM/GPRS): DL/
  - DL: Data Link
- L1(W / G): PHY (Physical Layer)
Video Streaming (RTSP)

**Video Streaming Server**
- Supports the video streaming service.
- MD8470A already provides a ‘Darwin Video Streaming Server’.

---

**Protocol Sequence**

**MS**
- TCP SYN & ACK (TCP)
  - Describe (RTSP)
  - Reply (RTSP)
  - Setup (RTSP)
  - Reply (RTSP)
  - Reply (RTSP)
  - Play (RTSP)
  - Reply (RTSP)
  - Payload (Content downloading : RTP)
  - Teardown (RTSP)
  - Reply (RTSP)
  - TCP FIN & ACK & RST (TCP)
  - PDP De-activation (Wireless Protocol)

**NW**
- TCP SYN & ACK (TCP)
  - Describe (RTSP)
  - Reply (RTSP)
  - Setup (RTSP)
  - Reply (RTSP)
  - Reply (RTSP)
  - Play (RTSP)
  - Reply (RTSP)
  - Payload (Content downloading : RTP)
  - Teardown (RTSP)
  - Reply (RTSP)
  - TCP FIN & ACK & RST (TCP)
  - PDP De-activation (Wireless Protocol)

---

All these parts are available in an external PC

File path (default contents) can be defined in the Streaming Server

PC inside of MD8470A

Port 'A' or Port 'B'

Port '0'

Discover What's Possible™
MD8470A-E-E-1
How to install the video streaming server and how to operate:
Please refer to ‘DarwinStreamingServerIntoMD8470A.doc’

How to operate video streaming demo on the MD8470A: Please refer to the demonstration note (section1-2)

*1: HTTP content download (Please refer the part 1 ‘HTTP’ in detail), at first UE accesses the default.htm content, then links to the streaming content in this example. (No.1 to No.5)

*2: When UE accesses the streaming content link, at first UE disconnects TCP and wireless connection (No.8)

*3: then activates TCP and wireless session again. (No.9 to No.12)
Video Streaming (RTSP)

Protocol Sequence (RTSP DESCRIBE)

1. RTSP
2. RTP/RTCP
3. TCP
4. UDP
5. IP
6. L3
7. L2 (Wireless Protocol)
8. L1 (Wireless Protocol)

*1: ‘URL’, we can confirm the URL (same as streaming Server setting), ‘DESCRIBE’ means requesting the detail content information. Server will reply to this request with SDP.

*2: ‘C Seq’ is the RTSP’s sequence number. (‘C Seq = 0’)

Video Streaming (RTSP)

Protocol Sequence (RTSP Reply)

1. RTSP
2. RTP/RTCP
3. TCP
4. UDP
5. IP
6. L3
7. L2 (Wireless Protocol)
8. L1 (Wireless Protocol)

*1: ‘200 OK’ means RTSP reply

*2: ‘C Seq’ is the RTSP’s sequence number. (‘C Seq = 0’)

we can confirm Streaming Server version application – type: sdp
Video Streaming (RTSP)

Protocol Sequence (RTSP Reply)

*1: There are 2 sessions. One is for video; the other is for voice. ‘Video’ is MP4 and ‘Track ID = 1’.

*2: This is for voice. ‘Voice’ is ‘AMR’, and ‘Track ID = 2’

---

Video Streaming (RTSP)

Protocol Sequence (RTSP SETUP)

*1: This is the ‘SETUP’ for the ‘Track ID = 2(voice)’.

*2: In the ‘SETUP’, Client informs the ‘available port number’. And ‘C Seq = 1’. And requests the Server side port number.
**Video Streaming (RTSP)**

### Protocol Sequence (RTSP Reply)

- **L1 (Wireless Protocol)**
- **L2 (Wireless Protocol)**
- **L3**
- **UDP**
- **TCP**
- **RTSP/RTCP**

*1* : This is the ‘Reply’ of ‘SETUP(Track ID = 2 : voice)’. ‘RTSP/1.0 200 OK’ means ‘Reply’

*2* : ‘C Seq = 1’, and Server informs the session number for the ‘Track ID = 2’, and also informs the port number (Server side : 6970-6971, Client 7000-7001).

---

**Video Streaming (RTSP)**

### Protocol Sequence (RTSP SETUP)

- **L1 (Wireless Protocol)**
- **L2 (Wireless Protocol)**
- **L3**
- **UDP**
- **TCP**
- **RTSP/RTCP**

*1* : This is the ‘SETUP’ for the ‘Track ID = 1(voice)’.

*2* : In the ‘SETUP’, Client informs the ‘available port number’. And ‘C Seq = 2’. And requests the Server side port number.
Video Streaming (RTSP)

Protocol Sequence (RTSP Reply)

- RTSP
- RTP/RTCP
- TCP
- UDP
- L3
- L2 (Wireless Protocol)
- L1 (Wireless Protocol)

*1: This is the ‘Reply’ of ‘SETUP(Track ID = 1 : video)’. ‘RTSP/1.0 200 OK’ means ‘Reply’

*2: ‘C Seq = 2’, and Server informs the session number for the ‘Track ID = 2’, and also informs the port number (Server side : 6970-6971, Client 7002-7003).

Video Streaming (RTSP)

Protocol Sequence (RTSP PLAY)

- RTSP
- RTP/RTCP
- TCP
- UDP
- L3
- L2 (Wireless Protocol)
- L1 (Wireless Protocol)

*1: ‘PLAY’ means ‘start the video streaming content’.

*2: ‘C Seq = 3’. And ‘PLAY’ with added session number
**Video Streaming (RTSP)**

**Protocol Sequence (RTSP Reply)**

- RTSP
- RTP/RTCP
- TCP
- UDP
- IP
- L3
- L2 (Wireless Protocol)
- L1 (Wireless Protocol)

*1: This is the ‘Reply’ of ‘PLAY’. ‘RTSP/1.0 200 OK’ means ‘Reply(Start the video streaming)’

*2: ‘C Seq = 3’, ‘PLAY’ with added session number and both ‘Track ID = 1’ and ‘Track ID = 2’.

---

**Video Streaming (RTSP)**

**Protocol Sequence (RTP Payload)**

- RTSP
- RTP/RTCP
- TCP
- UDP
- IP
- L3
- L2 (Wireless Protocol)
- L1 (Wireless Protocol)

*1: The video streaming content is downloaded with ‘RTP’ protocol.

*2: ‘frame 18’ means ‘Line No. 18 (SETUP for Track ID = 1 : video). You can confirm this (for Track ID = 1) with the port number information (7002).
Video Streaming (RTSP)

Protocol Sequence (RTP Payload)

*1 : The video streaming content is downloaded with ‘RTP’ protocol.

*2 : ‘frame 18’ means ‘Line No. 16 (SETUP for Track ID = 2 : video). You can confirm this (for Track ID = 2) with the port number information (7000).

*1 : sometimes there are some ‘Sender Reports’.

*2 : sometimes there are some ‘Receiver Reports’.
Video Streaming (RTSP)

**Protocol Sequence (RTSP TEARDOWN)**

RTSP → RTP/RTCP

TCP → UDP

L3 → IP

L2 (Wireless Protocol)

L1 (Wireless Protocol)

*1: ‘TEARDOWN’ means ‘Request to stop the video streaming’.

*2: ‘C Seq = 4’. And ‘TEARDOWN’ with added session number.

---

**Protocol Sequence (RTSP Reply)**

RTSP → RTP/RTCP

TCP → UDP

L3 → IP

L2 (Wireless Protocol)

L1 (Wireless Protocol)

*1: This is the ‘Reply’ of ‘TEARDOWN’. ‘RTSP/1.0 200 OK’ means ‘Reply(Stop video streaming)’

*2: ‘C Seq = 4’, ‘TEARDOWN’ with added session number.
Video Streaming (RTSP)

**Protocol Sequence (TCP FIN & Ack & RST)**

- RTSP
- RTP/RTCP
- TCP
- UDP
- L3
- L2 (Wireless Protocol)
- L1 (Wireless Protocol)

*Line940*: FIN means ‘request to disconnect’, and Ack means ‘Ack for previous TEARDOWN’. (UE to Server)

*Line941*: Retransmit of Line 939?

*Line942*: FIN means ‘request to disconnect’, and Ack means ‘Ack for previous ‘Reply of TEARDOWN’ (Server to UE)

*Line943,944*: Ack for previous TCP and RST (disconnect)

---

Part 3: Video Telephony
Video Telephony

3G H324M: 3G 324M consist of ‘H.245’ and ‘H.263/AMR’ (Multiplexing in H.223)
H.245: Protocol for the ‘control’ part of video telephony
H.263: Protocol for the ‘video’ part of the video telephony
AMR: Protocol for the ‘Voice’ part of the video telephony

L2(W-CDMA): MAC / RLC
- MAC: Media Access Control
- RLC: Radio Link Control

L2(GSM/GPRS): DL /
- DL: Data Link

L1(W / G): PHY (Physical Layer)

In this method, MD8470A sends through all 3G 324M data to ISDN terminal.

We can monitor H.245 log via Ethereal.
In this method, MD8470A has the following functions:

1. MD8470A’s H.245 sends a response for the command from MS (always waiting for MS command)
2. The content of MD8470A’s response message is just copying from the command message from MS
3. SRP is used in the case of ‘Command’; NSRP is used in the case of ‘Response’.

We can monitor H.245 log via Ethereal.

There are some more messages, but these 4 messages are important for connecting video Telephony.
Video Telephony

Protocol Sequence
(TerminalCapabilitySet)

MS
Signalling (Wireless Protocol)

NW

TerminalCapabilitySet (SRP Command)

(NSRP Response)

TerminalCapabilitySet (SRP Command)

(NSRP Response)

TerminalCapabilitySetAck (SRP Command)

(NSRP Response)

TerminalCapabilitySetAck (SRP Command)

(NSRP Response)

Protocol Sequence
(MasterSlaveDetermination)

MS
Signalling (Wireless Protocol)

NW

MasterSlaveDetermination (SRP Command)

(NSRP Response)

MasterSlaveDetermination (SRP Command)

(NSRP Response)

MasterSlaveDeterminationAck (SRP Command)

(NSRP Response)

MasterSlaveDeterminationAck (SRP Command)

(NSRP Response)
Video Telephony

Protocol Sequence (OpenLogicalChannel)

MS  Signalling (Wireless Protocol)  NW

OpenLogicalChannel (SRP Command)
(NSRP Response)

OpenLogicalChannel (SRP Command)
(NSRP Response)

OpenLogicalChannelAck (SRP Command)
(NSRP Response)

OpenLogicalChannelAck (SRP Command)
(NSRP Response)

There are for both ‘video’ and ‘Voice’ separately.

Protocol Sequence (MultiplexEntrySend)

MS  Signalling (Wireless Protocol)  NW

MultiplexEntrySend (SRP Command)
(NSRP Response)

MultiplexEntrySend (SRP Command)
(NSRP Response)

MultiplexEntrySendAck (SRP Command)
(NSRP Response)

MultiplexEntrySendAck (SRP Command)
(NSRP Response)
Part 4: SMS
(Single SMS, Concatenated SMS)

SMS (SMS – single SMS)

Protocol Stack

TP : Transfer Protocol
RP : Relay Protocol
CP : Control Protocol

L2(W-CDMA) : MAC / RLC
  MAC : Media Access Control
  RLC : Radio Link Control
L2(GSM/GPRS) : DL /
  DL : Data Link
L1(W / G) : PHY(Physical Layer)
**SMS (SMS – single SMS)**

**Protocol Stack**
- CP data is terminated in the network
- RP data is terminated in the SMSC (SMS center)
- TP data communicates between MSs

**Signaling Message (NAS message) structure**

<table>
<thead>
<tr>
<th>Num. Of Octets</th>
<th>bit no.</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>T flag + TI value</td>
<td>Protocol Discriminator</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Protocol Discriminator</td>
<td>CP-MT Type</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td></td>
<td>CP-UD length</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>n</td>
<td></td>
<td>Protocol discriminator</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>n</td>
<td></td>
<td>CP-UD length</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>n</td>
<td></td>
<td>CP-UD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**RP-UD structure**

<table>
<thead>
<tr>
<th>Num. Of Octets</th>
<th>bit no.</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 to 12</td>
<td>TP-DA</td>
<td>TP-DCS</td>
<td>TP-SCTS</td>
<td>TP-UDL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>TP-ID</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>TP-DCS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>TP-UDL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 to 140</td>
<td>TP-UD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**2 SMS receiving**

<table>
<thead>
<tr>
<th>Num. Of Octets</th>
<th>bit no.</th>
<th>7</th>
<th>6</th>
<th>5</th>
<th>4</th>
<th>3</th>
<th>2</th>
<th>1</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>TP-ID</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>TP-DCS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>TP-SCTS</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>TP-UDL</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 to 140</td>
<td>TP-UD</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Field Name**

<table>
<thead>
<tr>
<th>Field Name</th>
<th>supported in</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ti flag + TI value</td>
<td>Scenario</td>
</tr>
<tr>
<td>Protocol Discriminator</td>
<td>SMS Library</td>
</tr>
<tr>
<td>CP-MT Type</td>
<td>SMS Library</td>
</tr>
<tr>
<td>CP-UD Length</td>
<td>SMS Library</td>
</tr>
<tr>
<td>RP-MTI</td>
<td>SMS Library</td>
</tr>
<tr>
<td>RP-MR</td>
<td>SMS Library</td>
</tr>
<tr>
<td>RP-OA</td>
<td>Scenario</td>
</tr>
<tr>
<td>RP-DA</td>
<td>Scenario</td>
</tr>
<tr>
<td>RP-UDL</td>
<td>SMS Library</td>
</tr>
<tr>
<td>TP-DA</td>
<td>FakeSMSC</td>
</tr>
<tr>
<td>TP-ID</td>
<td>FakeSMSC</td>
</tr>
<tr>
<td>TP-DCS</td>
<td>FakeSMSC</td>
</tr>
<tr>
<td>TP-SCTS</td>
<td>FakeSMSC</td>
</tr>
<tr>
<td>TP-UDL</td>
<td>SMS Library</td>
</tr>
<tr>
<td>TP-UDH</td>
<td>FakeSMSC</td>
</tr>
<tr>
<td>TP-UD</td>
<td>SMS data</td>
</tr>
</tbody>
</table>

**Discover What’s Possible™**

MD8470A-E-E-1
**SMS (SMS – single SMS)**

**Protocol Sequence**

MS

TP

RP

CP

L3

L2 (Wireless Protocol)

L1 (Wireless Protocol)

---

**Protocol Sequence (sending SMS : TP)**

MS

TP

RP

CP

L3

L2 (Wireless Protocol)

L1 (Wireless Protocol)

NW

Signalling (Wireless Protocol)

SMS (sending SMS : TP)

CP-Ack (CP)

RP-Ack (RP)

CP-Ack (CP)

Signalling (Wireless Protocol)

---

*1 : we can confirm the CP layer data. We can see the CP layer data on the ‘NAS’ tag (RP, TP layer data on the ‘SMS’ tag)

*2 : we can see the raw RP layer data
### SMS (SMS – single SMS)

#### Protocol Sequence (sending SMS : TP)

- **TP**
- **RP**
- **CP**
- **L3**
- **L2(Wireless Protocol)**
- **L1(Wireless Protocol)**

**TP**

**RP**

**CP**

**L3**

**L2(Wireless Protocol)**

**L1(Wireless Protocol)**

**Protocol Sequence (sending SMS : TP)**

1. **TP Ack (TP)**
   - **TP**
   - **RP**
   - **CP**
   - **L3**
   - **L2(Wireless Protocol)**
   - **L1(Wireless Protocol)**

**TP Ack (TP)**

2. **CP Ack (CP)**
   - **TP**
   - **RP**
   - **CP**
   - **L3**
   - **L2(Wireless Protocol)**
   - **L1(Wireless Protocol)**

**CP Ack (CP)**

---

**1**: we can confirm the RP layer data (from NW to MS)

**2**: we can see the TP layer data (SMS Deliver, 7bit default alphabet, message class, SMS data itself, e.t.c.). Message class is from 0 to 3. 0 means ‘Display as soon as possible’, 1 means ‘Store the selected devices(SIM or memory)’, 2 means ‘Store the SIM’, 3 means ‘Forwarding to PC or PDA connected with MS.’

---

**1**: we can confirm the CP Ack data (from MS to NW). This is just CP layer(no RP, TP layer data) data.
**SMS (SMS – single SMS)**

### Protocol Sequence (RP Ack : RP)

1. **L1 (Wireless Protocol)**
2. **L2 (Wireless Protocol)**
3. **TP**
4. **RP**
5. **CP**

#### Protocol Sequence Details:

*1: we can confirm the RP raw data (from MS to NW).

*2: This is RP Ack data (from MS to NW).

---

**SMS (SMS – single SMS)**

### Protocol Sequence (CP Ack : CP)

1. **L1 (Wireless Protocol)**
2. **L2 (Wireless Protocol)**
3. **TP**
4. **RP**
5. **CP**

#### Protocol Sequence Details:

*1: we can confirm the CP Ack data for previous RP Ack (from NW to MS). This is just CP layer (no RP, TP layer data) data.
**SMS (SMS – Concatenated)**

**Protocol Sequence**

1. **TP**
2. **RP**
3. **CP**
4. **L3**
5. **L2 (Wireless Protocol)**
6. **L1 (Wireless Protocol)**

**Signalling (Wireless Protocol)**

1. SMS (sending SMS : TP)
2. CP-Ack(CP)
3. RP-Ack(RP)
4. CP-Ack(CP)

1. SMS (sending SMS : TP)
2. CP-Ack(CP)
3. RP-Ack(RP)
4. CP-Ack(CP)

*1 : we can confirm the CP layer data. We can see the CP layer data on the ‘NAS’ tag (RP, TP layer data on the ‘SMS’ tag)

*2 : we can see the raw RP layer data

---

**Discover What’s Possible™**

MD8470A-E-E-1

**Slide 87**

**Slide 88**
**Protocol Sequence (sending SMS : TP)**

*1: we can confirm the RP layer data (from NW to MS)

*2: TP layer data (SMS Deliver, 7bit default alphabet, message class, etc.)

*3: this is not last message (concatenated massage) this TP-UD has TP-UDH (User Data Header). TP-UD ‘050003000201…’ means ‘05’ is the length of UDH, next ‘00’ means IEI (in the case of concatenate SMS), next ‘03’ means the length of this IEI, next ‘00’ means sms sequence number, next ‘02’ means ‘how many concatenate’ next ‘01’ means 1st block of concatenated SMS.

**Protocol Sequence (CP Ack : CP)**

*1: we can confirm the CP Ack data (from MS to NW). This is just CP layer (no RP, TP layer data) data.
**SMS (SMS – Concatenated)**

### Protocol Sequence (RP Ack : RP)

- **TP**
- **RP**
- **CP**
- **L3**
- **L2** (Wireless Protocol)
- **L1** (Wireless Protocol)

#### *1*
- we can confirm the RP raw data (from MS to NW).

#### *2*
- This is RP Ack data (from MS to NW)

---

**SMS (SMS – Concatenated)**

### Protocol Sequence (CP Ack : CP)

- **TP**
- **RP**
- **CP**
- **L3**
- **L2** (Wireless Protocol)
- **L1** (Wireless Protocol)

#### *1*
- we can confirm the CP Ack data for previous RP Ack (from NW to MS). This is just CP layer (no RP, TP layer data) data.
**SMS (SMS – Concatenated)**

### Protocol Sequence

- **L1 (Wireless Protocol)**
- **L2 (Wireless Protocol)**
- **L3**

**TP**

- **RP**

- **CP**

**Protocol Sequence (sending SMS : TP)**

*1: we can confirm the CP layer data. We can see the CP layer data on the ‘NAS’ tag (RP, TP layer data on the ‘SMS’ tag)

*2: we can see the raw RP layer data

---

**SMS (SMS – Concatenated)**

### Protocol Sequence

- **L1 (Wireless Protocol)**
- **L2 (Wireless Protocol)**
- **L3**

**TP**

- **RP**

- **CP**

**Protocol Sequence (sending SMS : TP)**

*1: we can confirm the RP layer data (from NW to MS)

*2: TP layer data (SMS Deliver, 7bit default alphabet, message class, e.t.c.)

*3: this is the last message (concatenated message) this TP-UD has TP-UDH (User Data Header). TP-UD ‘050003000202…’ means ‘005’ is the length of UDH, next ‘00’ means IEI (in the case of concatenate SMS), next ‘03’ means the length of this IEI, next ‘00’ means sms sequence number, next ‘02’ means ‘how many concatenate’ next ‘02’ means 2nd block of concatenated SMS.
**SMS (SMS – Concatenated)**

**Protocol Sequence (CP Ack : CP)**

- TP
- RP
- L3
- L2 (Wireless Protocol)
- L1 (Wireless Protocol)

*1: we can confirm the CP Ack data (from MS to NW). This is just CP layer (no RP, TP layer data) data.

**Protocol Sequence (RP Ack : RP)**

- TP
- RP
- L3
- L2 (Wireless Protocol)
- L1 (Wireless Protocol)

*1: we can confirm the RP raw data (from MS to NW).

*2: This is RP Ack data (from MS to NW)
*1 : we can confirm the CP Ack data for previous RP Ack (from NW to MS). This is just CP layer (no RP, TP layer data) data.

Part 5: MMS
(HTTP-Submit, HTTP-Retrieve,
WAP-Submit, WAP-Retrieve)
MMS (System Architecture)

MD8470A
FakeSMSC
(MMSC Simulator)

Simulation Control
Scenario / WNS

Mobile Network Simulation

MMSC / PPG

User’s Test Server or
Third Party Server
Application

WAP Gateway
(Kannel)

MMS File

SMS File

Air Link

MMS (MMS – HTTP)

System Architecture

- SMS data
- IP Packet Data (HTTP)
- Retrieve (MMS to MS)
- Submit (MMS to MMSC)

Retrieving Procedure:
1. Sending notification (SMS)
2. Access MMSC (IP-HTTP)
3. Downloading MMS (IP-HTTP)

* : SMS session is used only in the case of retrieving

Submitting case:
1. Sending MMS (IP-HTTP)
MMS Submit (MMS – HTTP)

Protocol Sequence

MS

PDP Activation (Wireless Protocol)

TCP SYN & ACK (TCP)

M-send-req (MMSE)

Sending MMS contents (MMSE: many times)

M-send-conf (MMSE)

TCP FIN & ACK & RST (TCP)

PDP De-Activation (Wireless Protocol)

NW

BTS/NW

MMSC

SMSC

MS

*1: These processes are the same as the part ‘HTTP content download (Please refer to part 1 ‘HTTP’ in detail)’.
**MMS Submit (MMS – HTTP)**

**Protocol Sequence (m-send-req)**

*1: ‘POST’ means ‘sending from MS to Server’. ‘xst=test’ means ‘UserID=xst, Password=test’ for MMSC server. ‘HOST’ is ‘192.168.1.2(IP address), 7082(port number).

*2: Message Type = ‘m-send-req (sending MMS)’ And we can confirm ‘Class’, ‘Priority’, ‘deliver report’, ‘read report’, e.t.c...

---

**MMS Submit (MMS – HTTP)**

**Protocol Sequence (m-send-req)**

*1: ‘Data’ means MMS data itself. MMS data is big, so MS send MMS data divided many blocks. This is the first block of MMS data (not completed yet).
*1: ‘Continuation or non-HTTP traffic’ means ‘MMS Data sending (from MS to Server)’. We can confirm the TCP Ack (from Server to MS) for each data. (Server side is smooth enough, so we can confirm the Ack for each data.)

*2: ‘200 OK’ means ‘succeed of sending data at HTTP level.

*2: ‘m-send-conf’ means the acknowledge from Server to MS for the success of MMS sending.
*1: These processes are almost the same as the part ‘HTTP content download (Please refer the part 1 ‘HTTP’ in detail)’. No.28: Ack for No.27 and disconnect request from Server to MS. No.29 is Ack for No.28. No.30 is disconnect request from MS to Server. No.31 is Ack for No.30.

MMS Submit (MMS – HTTP)

MMS Retrieve (MMS – HTTP)
Please refer to the ‘mms-http-retrieve.lgx’ file. We can confirm 2 concatenated SMS procedure. The detail of this concatenated SMS, please refer the Part3 ‘Concatenated SMS’. TP-UDH is ‘0B 00 03 00 02 02 05 04 0B 84 23 F0’. First ‘0B’ means total TP-UDH length. Next ‘00 03 00 02’ is 2nd block of 2 concatenated SMS. Last ‘05 04 0B 84 23 F0’ means ‘05’ -> IEL of this information, ‘04’ means this length, ‘0B 84’ means the port number(2948) for destination (MS side), ‘23 F0’ means the port number(9200) for source (Server side) The port number 2948 is used for ‘WAP push’. ‘WAP push’ technology is used in this notification.

*1 : These process is same as the part ‘HTTP content download (Please refer to part 1 ‘HTTP’ in detail’).
*1 : We can confirm the ‘URI(path of MMS content)’, Host. ‘GET’ means the ‘request for downloading MMS’.

*1 : HTTP 200 OK (OK for the request of MMS content download)

*2 : we can confirm ‘m-retrieve-conf’ message, ‘Subject’, ‘From’, ‘Deliver Report’, ‘Read Report’, ‘Priority’, e.t.c.. ‘Data’ field is MMS content data itself (divided many blocks and sending: not completed yet)
MMS Retrieve (MMS – HTTP)

Protocol Sequence (downloading MMS)

*1: ‘Continuation or non-HTTP traffic’ means ‘MMS Data downloading (from Server to MS)’. We can confirm the TCP Ack (from MS to Server) for each 3 data. (MS side is not smooth enough, so we can confirm the Ack for each 3 data. But this is just our guess of this behavior.)

MMS Retrieve (MMS – HTTP)

Protocol Sequence (TCP FIN, Ack & RST)

*1: These process is almost same as the part ‘HTTP content download (Please refer the part 1 ‘HTTP’ in detail)’. No.27 is Ack for No.26. No.28 is disconnect request from Server to MS and Ack for No.27. No.29 is Ack for No.28. No.30 is disconnect request from MS to Server. No.31 is Ack for No.30. No.32 is disconnect.
MMS Retrieve (MMS – HTTP)

**Protocol Sequence**

(TCP SYN & Ack)

1. **MS** → **BTS/NW**: MMS Retrieve (MMS – HTTP)
   - **SMSC**: Sends an HTTP request
   - **MMSC**: Receives the request

*1: These process is same as the part ‘HTTP content download (Please refer to part 1 ‘HTTP’ in detail’).

**Protocol Sequence**

(m-notifyresp-ind)

1. **MS** → **BTS/NW**: MMS Retrieve (MMS – HTTP)
   - **SMSC**: Sends an HTTP request
   - **MMSC**: Receives the request

   ‘HOST’ is ‘192.168.1.2(IP address), 7082(port number).

*2: Message Type = ‘m-notifyresp-ind (Ack of Notification(SMS))’ And we can confirm ‘Status(Retrieved)’, e.t.c.
MMS Retrieve (MMS – HTTP)

Protocol Sequence (HTTP OK)

*1 : HTTP 200 OK (OK for the ‘m-notifyresp-ind’ (POST : HTTP layer) message)

Protocol Sequence (TCP FIN, Ack & RST)

*1 : These process is almost the same as the part ‘HTTP content download (Please refer to part 1 ‘HTTP’ in detail’).
No.38 is disconnect request from Server to MS. No.39 is Ack for No.38 and disconnect request from MS to Server. No.40 is Ack for No.39. No.41 is disconnect.
MMS (MMS – WAP)

System Architecture

- -----
  - SMS data
  - IP Packet Data (HTTP)
  - Retrieve (MMS to MS)
  - Submit (MMS to MMSC)

Retrieving Procedure:
1. Sending notification (SMS)
2. Access MMSC (IP-HTTP)
3. Downloading MMS (IP-HTTP)

*: SMS session is used only in the case of retrieving

Submitting case:
1. Sending MMS (IP-HTTP)

MMS Submit (MMS – WAP)

Protocol Sequence

- PDP Activation (Wireless Protocol)
- WSP Connection (WTP+WSP)
- WTP Invoke (WTP)
- Sending MMS contents (MMSE: many times)
- M-send-req (MMSE)
- M-send-conf (MMSE)
- WSP Disconnect (WSP)
- PDP De-Activation (Wireless Protocol)
MMS Submit (MMS – WAP)

Protocol Sequence (WSP Connection)

1: No.1 is ‘WSP Connect’ request (from MS to Server).
No.2 is ‘WSP ConnectReply’ (from Server to MS).
No.3 is Ack for No.2. (Please refer to part 1 ‘WAP content download’ in detail.)

MMS Submit (MMS – WAP)

Protocol Sequence (WTP Invoke (WTP))

1: PDU Type: ‘Invoke(request to upper layer)’. This data is not last packet (This is the 1st block of the sending MMS data).
2: ‘Payload’ means ‘sending MMS data’. The detail is described in No.16 (reassembled in : 16).
**MMS Submit (MMS – WAP)**

**Protocol Sequence (Sending MMS contents)**

*1: ‘Segmented Invoke’ means ‘MMS Data sending (from MS to Server)’. We can confirm the WTP Ack (from Server to MS) for once in each 3 data. (‘How often is this Ack’ is depending on the MS.)

*2: ‘segmented Invoke (8)’ is the last packet of MMS data.

---

**MMS Submit (MMS – WAP)**

**Protocol Sequence (m-send-req (MMSE))**

*1: we can confirm the result for sending MMS data in each data block.

*2: PDU Type : ‘Post (sending from MS to Server)’ And we can confirm the URI (Http://192.168.1.2:7082/xst=test).
MMS Submit (MMS – WAP)

Protocol Sequence (m-send-req (MMSE))

1: Message Type = 'm-send-req (sending MMS)' And we can confirm ‘Class’, ‘Priority’, ‘deliver report’, ‘read report’, e.t.c. In the case of ‘MMS-Submit(HTTP)’, at first ‘m-send-req’ then ‘MMS data’ itself, but in the case of ‘MMS-Submit(WAP)’, at first sending ‘MMS data’ itself (segmented Invoke), then ‘m-send-req’.

2: we can confirm the MMS data in detail here.

MMS Submit (MMS – WAP)

Protocol Sequence (m-send-conf (MMSE))

1: PDU Type : ‘Reply’ is response for ‘POST(WSP)’.

2: Message-Type : ‘m-send-conf’ is Ack for ‘m-send-req’
**MMS Submit (MMS – WAP)**

**Protocol Sequence**
(WSP Disconnect)

*1: No.18 is Ack for No.17 (WTP level). No.19 is WSP disconnect.

**MMS Retrieve (MMS – WAP)**

**Protocol Sequence**

---

Discover What's Possible™
MD8470A-E-E-1
Slide 127

---

Discover What's Possible™
MD8470A-E-E-1
Slide 128

---

64
Please refer to ‘mms-wap-retrieve.lgx’ file. We can confirm a single SMS procedure. The detail of this concatenated SMS, please refer the Part3 ‘Single SMS’. TP-UDH is ’06 05 04 0B 84 23 F0’. First ‘06’ means total TP-UDH length. Next ’05 04 0B 84 23 F0’ means ’05’ -> IELI of this information, ’04’ means this length, ’0B 84’ means the port number(2948) for destination (MS side), ’23 F0’ means the port number(9200) for source (Server side) The port number 2948 is used for ‘WAP push’. ‘WAP push’ technology is used in this notification.

*1 : No.1 is ‘WSP Connect’ request (from MS to Server). No.4 is Ack for No.1. No.5 is Ack for No.4. No.6 is ‘WSP ConnectReply’ (from Server to MS). (Please refer to part 1 ‘WAP content download’ in detail.)
MMS Retrieve (MMS – WAP)

Protocol Sequence (GET (WSP))

*1 : First WTP’s PDU Type is Ack for No.6(WSP Connect reply). Second WTP’s PDU Type is Invoke (request for upper layer).

*1 : PDU Type : GET (request for downloading MMS content) We can confirm URI.
MMS Retrieve (MMS – WAP)

Protocol Sequence (Downloading MMS)

*1: ‘Segmented Result’ is reply for previous Invoke (WSP layer: Get). ‘Payload’ means ‘MMS Data downloading (from Server to MS)’. We can confirm the WTP Ack(from MS to Server) for once in each 3 data. (*How often is this Ack* is depending on the MS.) This is not last Packet. ‘MMS downloading’ is still going on.

MMS Retrieve (MMS – WAP)

Protocol Sequence (m-retrieve-conf(-MMSE))

*1: ‘Segmented Result’ is reply for previous Invoke (WSP layer: Get). This is the last packet of ‘MMS downloading’.

*2: We can confirm the downloaded MMS result.
Protocol Sequence (m-retrieve-conf)

1: PDU Type: ‘Reply’ is response for ‘GET(WSP)’. 
2: We can confirm ‘m-retrieve-conf’ message. And also can confirm ‘Subject’, ‘From’, ‘Deliver Report’, ‘Read Report’, ‘Priority’, e.t.c. . ‘Data’ field is MMS content data itself. In the case of ‘MMS-Retrieve(HTTP)’, at first ‘m-retrieve-conf’ then ‘MMS data’ itself, but in the case of ‘MMS-Retrieve(WAP)’, at first sending ‘MMS data’ itself (WSP-GET), then ‘m-retrieve-conf’.

Protocol Sequence (m-notifyresp-ind)

1: This WTP Ack is for ‘m-retrieve-conf’.(WTP layer).
2: PDU Type: POST (from MS to Server) We can confirm the URI .
3: Message Type = ‘m-notifyresp-ind (Ack of Notification(SMS))’ And we can confirm ‘Status(Retrieved)’, e.t.c. .
MMS Retrieve (MMS – WAP)

**Protocol Sequence**
(WSP OK)

1. **MMSC**
2. **SMSC**
3. **WAP Gateway**
4. **MS**
5. **BTS/NW**

*1: WSP 200 OK (OK for the ‘m-notifyresp-ind’ (POST: WSP layer) message)

---

MMS Retrieve (MMS – WAP)

**Protocol Sequence**
(WSP Disconnect)

1. **MMSC**
2. **SMSC**
3. **WAP Gateway**
4. **MS**
5. **BTS/NW**

*1: No.57 is Ack for No.56 (WTP Layer). No.58 is WSP disconnect.
Part 6: DRM

DRM (What is DRM?)

- DRM is Digital Rights Management
- The OMA (Open Mobile Alliance) promotes standardization
- DRM consists of ‘Content’ and a ‘Rights object’
- A ‘Rights object’ contains information such as ‘times for using’ and ‘term (how may days) for using’
DRM (What is DRM?)

- There are 3 DRM methods:
  - Forward lock
    - This DRM contents supports only ‘cannot forward’.
  - Combined delivery lock
    - Server side replies with ‘content’ and ‘Rights object’ at the same time to MS in this method. A ‘Rights object’ supports ‘cannot forward’, ‘times for displaying/playing/executing/printing out’ and ‘How many days (absolute/relative expression) for displaying/playing/executing/printing out’.
  - Separate delivery lock
    - Server side replies with ‘content’ and ‘Rights object’ separately(*1). A function of a ‘Rights object’ has is same as ‘Combined delivery lock’. Available for copying only content (‘Rights object is not allowed to forward’).

(*1) : There are 2 ways for downloading ‘Rights Object’. One is downloading on the Packet session; the other is downloading on the SMS session (WAP Push).

---

DRM (System Architecture)

- System Architecture is the same as for ‘packet’. The difference is only the contents itself.
- Just by preparing the DRM contents and copying it to the MD8470A, we can test DRM.
DRM (Forward Lock)

- Protocol Sequence for downloading ‘Forward Lock’ content is same as ‘Packet communication (HTTP, WAP)’ (Please refer to Part1 in detail)

- Please confirm downloaded content is ‘Forward Lock’ content and cannot copy/forward

---

**DRM (Forward Lock)**

**Protocol Sequence**

**MS**

1. Signalling (Wireless Protocol)
2. TCP Activation
3. GET (HTTP)
4. OK (HTTP)
5. Downloading contents (many times)
6. TCP de-activation
7. PDP de-activation (Wireless Protocol)

**NW**
**DRM (Forward Lock)**

**TCP Activation & Get (HTTP)**

*1: We can confirm the TCP connection process. (we can also confirm the .html content access for welcome pages.) In No.12, we can confirm the access for DRM content and file type is `.dm`.

Impossible to copy to external memory.

Impossible to attach to mail.

Available for playing.

---

**DRM (Forward Lock)**

**OK (HTTP)**

*1: We can confirm the downloading DRM content. We can also confirm the `Content-Type` is DRM.

Impossible to copy to external memory.

Impossible to attach to mail.

Available for playing.
**DRM (Forward Lock)**

**Downloading & TCP De-activation**

*1 : We can confirm the downloading DRM content. After downloading, we can also confirm the TCP disconnection process.

* : Then please confirm the downloaded content is ‘DRM (Forward Lock)’ by checking the detail content information.

**Impossible to copy to external memory.**

**Impossible to attach to mail.**

**Available for playing.**

---

**DRM (Combined Delivery Lock)**

- Protocol Sequence for downloading ‘Combined Delivery Lock’ content is same as ‘Packet communication(HTTP, WAP)’ (Please refer to Part1 in detail)

- Please confirm downloaded content is ‘Combined Delivery Lock’ content and cannot copy/forward and check the limit of ‘How many times’ and ‘until when’
**DRM (Combined Delivery Lock)**

### Protocol Sequence

**MS**

- Impossible to attach to mail
- Impossible to copy to external memory
- Not available for playing after expiration

**NW**

- Signalling (Wireless Protocol)
- TCP Activation
- GET (HTTP)
- OK (HTTP)
- Downloading contents (many times)
- TCP de-activation
- PDP de-activation (Wireless Protocol)

---

**TCP Activation & Get (HTTP)**

**MS**

- Impossible to attach to mail
- Impossible to copy to external memory
- Not available for playing after expiration

**Sample: Content**

*1: We can confirm the TCP connection process. (we can also confirm the .html content access for welcome pages.) In No.12, we can confirm the access for DRM content and file type is `.dm`.

---

*Anritsu*

MD8470A-E-E-1
**DRM (Combined Delivery Lock)**

**OK (HTTP)**

1. Impossible to copy to external memory.
2. Impossible to attach to mail.
3. Not available for playing after expiration.

*1: We can confirm the downloading DRM content. We can also confirm the ‘Content-Type’ is DRM.

**Availability for playing** *1

---

**DRM (Combined Delivery Lock)**

**Downloading & TCP De-activation**

1. Impossible to copy to external memory.
2. Impossible to attach to mail.
3. Not available for playing after expiration.

*1: We can confirm the downloading DRM content. After downloading, we can also confirm the TCP disconnection process.

* : Then please confirm the downloaded content is ‘DRM (Combined Delivery Lock)’ by checking the detail content information (please check ‘how many times for playing’, etc.).
DRM (Separate Delivery Lock)

- Protocol Sequence for downloading ‘Separate Delivery Lock’ content is same as ‘Packet communication (HTTP, WAP)’ (Please refer to Part1 in detail). Please confirm 2 access (downloading), one is for ‘downloading content’, the other is for ‘downloading Rights object’.

- Please confirm downloaded content is ‘Separate Delivery Lock’ content and can not copy/forward and check the limit of ‘How many times’ and ‘until when’. And please confirm that you can copy the content, but cannot copy the ‘Rights Object’.

**Protocol Sequence**

1. MS
2. HTTP download
3. Not-Available for playing
4. Impossible to attach to mail
5. Impossible to copy to external memory
6. Rights
7. Accesses
8. WAP Push
9. Available for playing
10. Not available for playing after expiration

**MS**

- TCP Activation
- GET (HTTP)
- OK (HTTP)
- Downloading contents (many times)
- TCP de-activation
- TCP Activation
- GET (HTTP)
- OK (HTTP)
- Downloading ‘Rights Object’
- TCP de-activation
- PDP de-activation (Wireless Protocol)
DRM (Separate Delivery Lock)

**TCP Activation & Get (HTTP)**

*1: We can confirm the TCP connection process. (we can also confirm the .html content access for welcome pages.) In No.12, we can confirm the access for DRM content and file type is `.dcf`.

**OK (HTTP)**

*1: We can confirm the downloading DRM content. We can also confirm the ‘Content-Type’ is DRM.
**DRM (Separate Delivery Lock)**

### Downloading & TCP De-activation

- **MS**
  - Not-Available for playing
  - Impossible to copy to external memory.
  - Impossible to attach to mail.

- **HTTP download**
  - Rights

- **Available for playing**

*1: We can confirm the downloading DRM content. After downloading, we can also confirm the TCP disconnection process.

*: Then please confirm the downloaded content can not open, because there is no ‘Rights object’.

### TCP Activation & Get (HTTP)

- **MS**
  - Not-Available for playing
  - Impossible to copy to external memory.

- **HTTP download**

*1: We can confirm the TCP connection process.

(we can also confirm the .html content access for welcome pages.) In No.77, we can confirm the access for DRM Rights object and file type is ‘.dr’.

---

Discover What’s Possible™
MD8470A-E-E-1

---

79
**DRM (Separate Delivery Lock)**

**OK (HTTP)**

*1: We can confirm the downloading DRM Rights object. We can also confirm the ‘Content-Type’ is ‘Rights+xml’. Rights object is small enough for downloading at once, so there is no more packet in the trace log.

**Right (HTTP)**

*1: TCP De-activation

*1: We can confirm the TCP disconnection process, after Rights object downloading.

* : Then please confirm the downloaded content is ‘DRM (Separate Delivery Lock)’ by checking the detail content information (please check ‘how many times for playing’, etc). And please confirm you can copy the content (but you cannot copy the ‘Rights object’).
Specifications are subject to change without notice.