NowSMS MMSC Training: Phone to MMSC Protocol (MM1)

NOVEMBER 2015

MMS Protocols

Overall Architecture: 3GPP TS 23.140
http://www.3gpp.org

Over-the-Air Phone to MMSC: Open Mobile Alliance (OMA) – MMS Encapsulation Protocol
http://www.openmobilealliance.org

External Connectivity Protocols: 3GPP TS 23.140

- MM4 – Interoperator
- MM7 – Value Added Service Provider

MM1 Overview

MM1 is the over-the-air protocol for MMS message transmission between a phone and the network.

Defined by Open Mobile Alliance (OMA) – MMS Encapsulation Protocol
http://www.openmobilealliance.org

Based on HTTP, WAP Push and MIME technologies

Multimedia Messaging Service (MMS)

Designed to compliment SMS by providing support for multimedia content

- Text
- Images
- Video
- Audio
- Contact Objects
- Calendar Objects
- SMIL Presentation (legacy support)

MMSC (MMS Centre)

Provides store-and-forward MMS Messaging Service for mobile phone subscribers

Temporarily stores MMS messages awaiting subscriber retrieval

May interconnect with MMSCs for other network operators, either with direct connections, or via an aggregator

May convert MMS messages to other formats (e.g., SMS with web link) for non-subscribers or external recipients

May allow MMS submissions from Value Added Service Providers

May apply content adaptation on message content based upon capabilities of the receiving device

MMS and Group Messaging

Supports multiple recipients and group conversations

The biggest driver of MMS message traffic growth has been its use in modern smartphones for group messaging

Even when the content is text only, MMS is preferred for group messaging because recipients can reply all, allowing all group members to see all messages.

MM1 Protocol Data Units (PDUs)

PDUs are the data elements that are exchanged between a mobile phone and the MMSC

PDUs are encoded using the MIME type application/vnd.wap.mms-message

PDUs are transferred as the payload of HTTP requests, HTTP responses, or WAP push messages

Example (HTTP POST):

```
POST / HTTP/1.1
Content-Type: application/vnd.wap.mms-message
Content-Length: 99999
```

MM1 PDU Kinds

There are three kinds of MM1 PDUs:

1. **Request**: Denoted as type-name.req
2. **Confirmation** (Response): Denoted as type-name.conf
3. **Indication** (notification): Denoted as type-name.ind (not confirmed)

MM1 PDU Transmission

When the client needs to send a PDU (Request or Indication) to the MMSC, it always uses HTTP POST.

- If the MMSC receives a Request PDU via HTTP, it generates an appropriate Confirmation PDU in the HTTP response.
- If the MMSC receives an Indication PDU via HTTP, it generates an empty HTTP OK response (status code 200 or 204).

When the MMSC needs to send a PDU (Indication only) to a client, it only uses WAP Push.

WAP 1.x Client Considerations

Older MMS clients based upon the WAP 1.x protocol use the Wireless Session Protocol (WSP) instead of HTTP.

These clients must use a WAP Gateway which converts between WSP and HTTP.

The MM1 interface at the MMSC is always HTTP based.

MM1 PDUs

Transaction	PDU Type Name	Originated By	Transport
Send Message	m-send.req (.conf)	Client	HTTP POST
New Message Received Notification	m-notification.ind	MMSC	WAP PUSH
Acknowledge New Message Received Notification (optional)	m-notifyresp.ind	Client	HTTP POST
Retrieve Message	HTTP GET / m-retrieve.conf	Client	HTTP GET
Acknowledge Retrieve Message Complete (optional)	m-acknowledge.ind	Client	HTTP POST
Delivery Report	m-delivery.ind	MMSC	WAP PUSH
Read Report from Client	m-read-rec.ind	Client	HTTP POST
Read Report to Client	m-read-orig.ind	MMSC	WAP PUSH
Forward Message	m-forward.req (.conf)	Client	HTTP POST

MMS Message Structure

An MMS Message has a similar structure to an SMTP E-Mail Message

Structure based upon SMTP and Multipart Internet Mail Extensions (MIME)

- Envelope – Recipients for in-transit message instance.
- Headers – Sender, Displayed Recipients, Subject, other attributes
- Content – MIME encoded multipart content. One or more multimedia objects (text, image, video, etc.)

MMS Envelope

Routing Attributes (sender and recipient) for an instance of an in-transit message.

Example: A message is sent to multiple recipients, some of which are subscribers of a different mobile network. The message may be split by the MMSC for delivery to another MMSC. The MMS envelope for the message instance sent to the other MMSC will be only the remote recipients. The envelope list of recipients tells the other MMSC which recipients to deliver the message to.

The MMS Header will still contain a list of all recipients to provide full group messaging support.

In MM4/SMTP these are represented by MAIL FROM: / RCPT TO: commands.

MMS Headers

Sender

Recipients (To/CC/BCC)

Subject

Priority

Message Class (Personal, Advertisement)

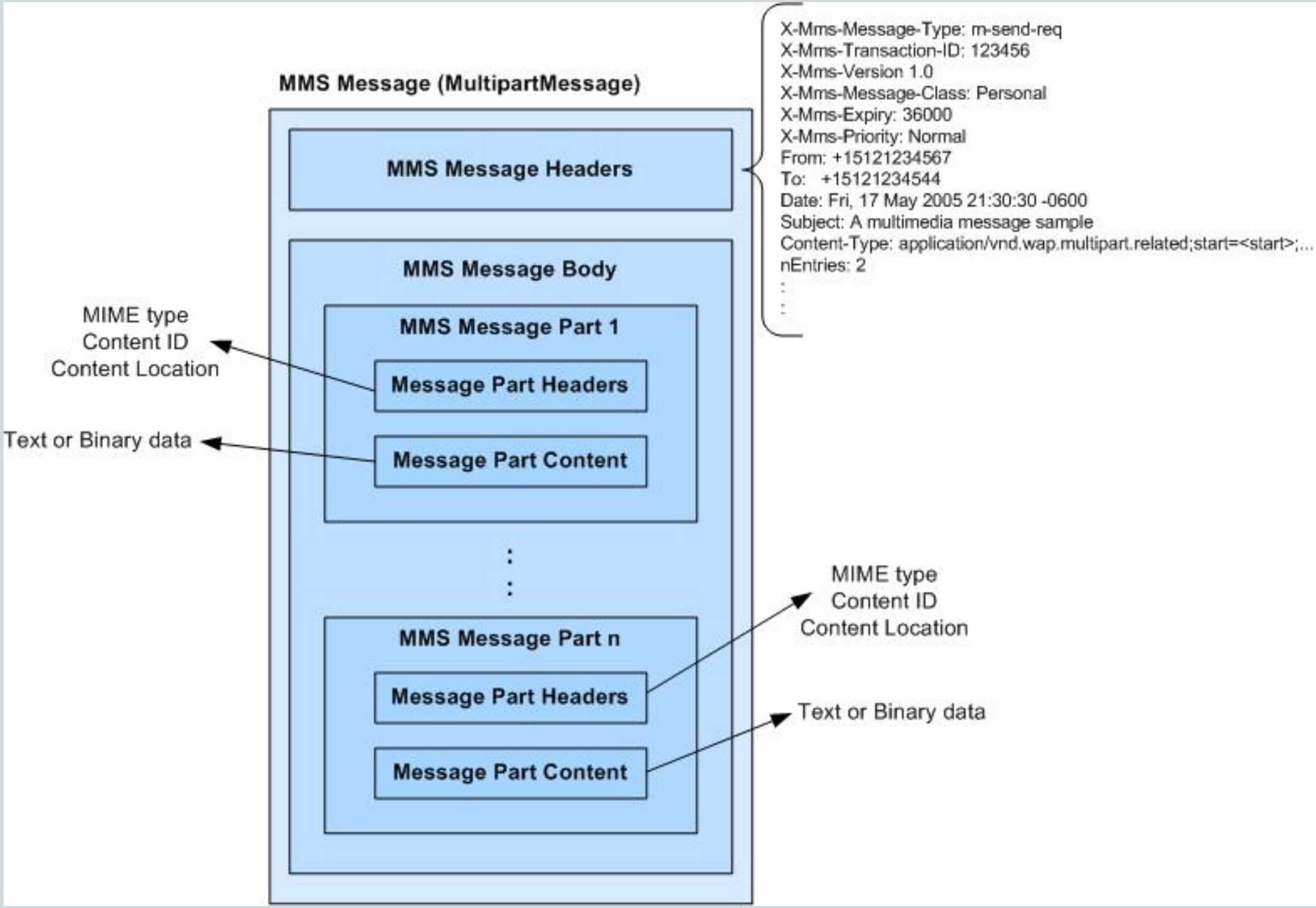
Message Type (can be delivery report or read report)

Message ID

MMS Content

Multipart (MIME) object containing one or more of the following:

- Text
- Images
- Video
- Audio
- Contact Objects
- Calendar Objects
- SMIL Presentation (legacy support)


MMS Message (MultipartMessage)
MMS Message Headers
MMS Message Body
MMS Message Part 1
Message Part Headers
Message Part Content
MMS Message Part n
Message Part Headers
Message Part Content
MIME type
Content ID
Content Location
Text or Binary data
MIME type
Content ID
Content Location
Text or Binary data
X-Mms-Message-Type: m-send-req
X-Mms-Transaction-ID: 123456
X-Mms-Version 1.0
X-Mms-Message-Class: Personal
X-Mms-Expiry: 36000
X-Mms-Priority: Normal
From: +15121234567
To: +15121234544
Date: Fri, 17 May 2005 21:30:30 -0600
Subject: A multimedia message sample
Content-Type: application/vnd.wap.multipart.related;start=<start>;...
nEntries: 2

MM1 Message Encoding

No MMS Envelope: All recipients are in MMS Headers

The MMS Encapsulation Protocol defines the MIME type application/vnd.wap.mms-message, which consists of headers and an optional content body.

MMS Headers are encoded in a binary format, following the WAP Session Protocol (WSP)

- WSP defines a single byte code for common header names and values to reduce message size
- X-MMS-Message-Type: m-retrieve.conf reduces from a 36 character string to two bytes: 8C 84

MM1 Message Encoding

MMS Content is encoded as multipart MIME object.

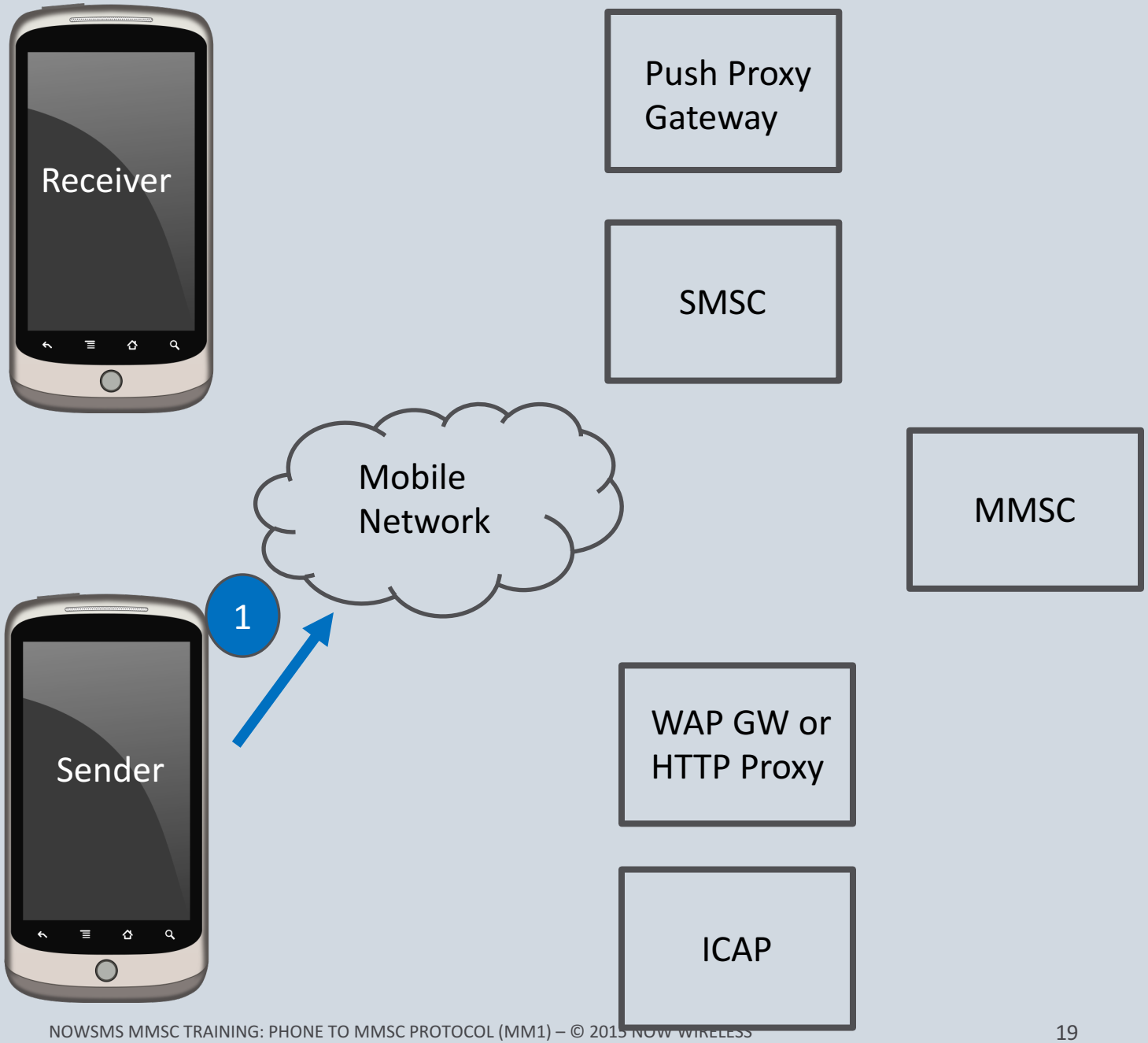
The multipart object is are encoded in a binary format, following the WAP Session Protocol (WSP)

- application/vnd.wap.multipart.related is used in place of multipart/related
- application/vnd.wap.multipart.mixed is used in place of multipart/mixed

MM1 Transaction: Step 1

Sender activates data connection to MMS APN

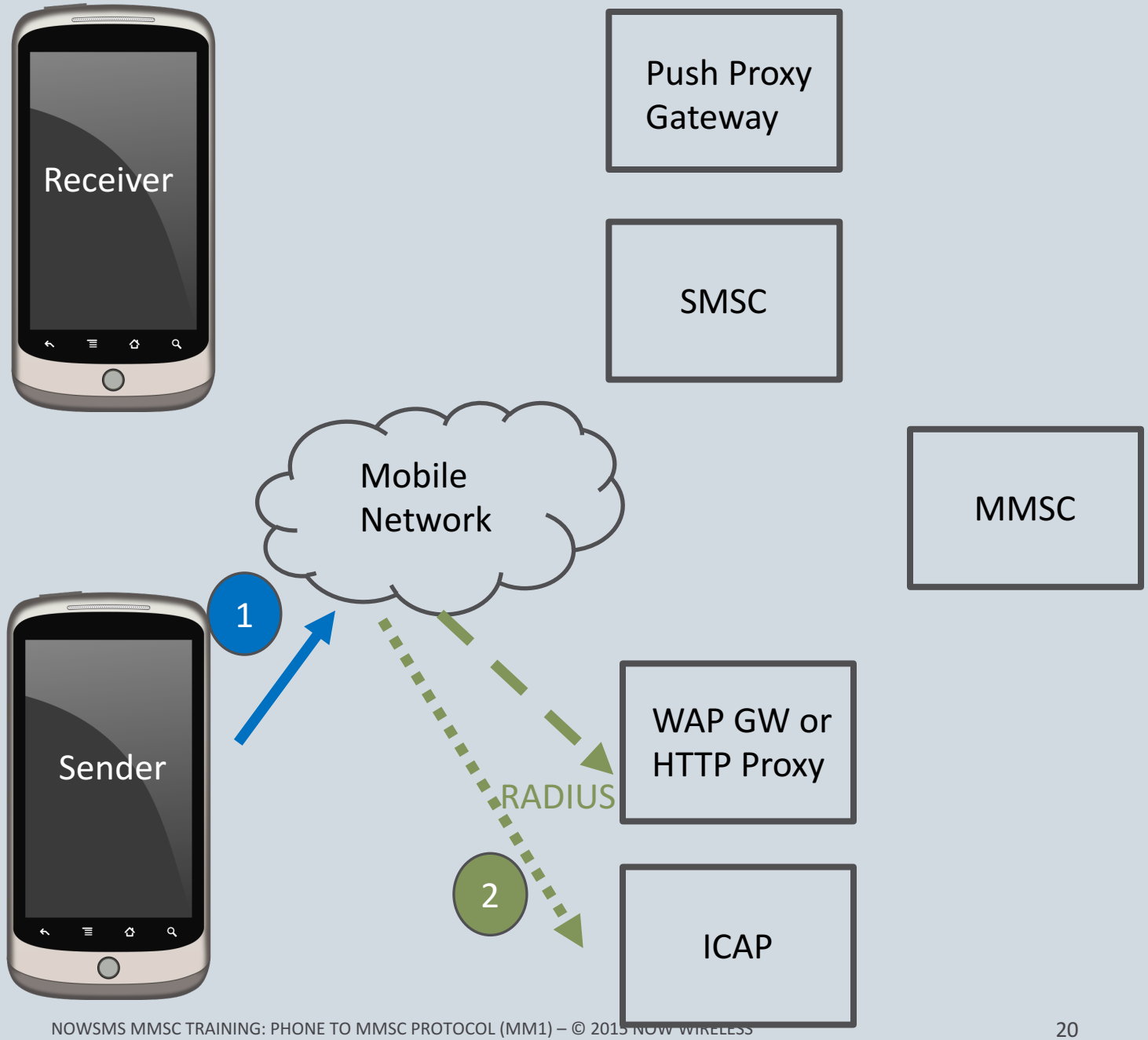
(APN setting configured in the phone)



MM1 Transaction: Step 2

GGSN sends RADIUS accounting message to notify the network about the device IP assignment.

WAP Gateway or ICAP server maintains table of active device IP addresses.



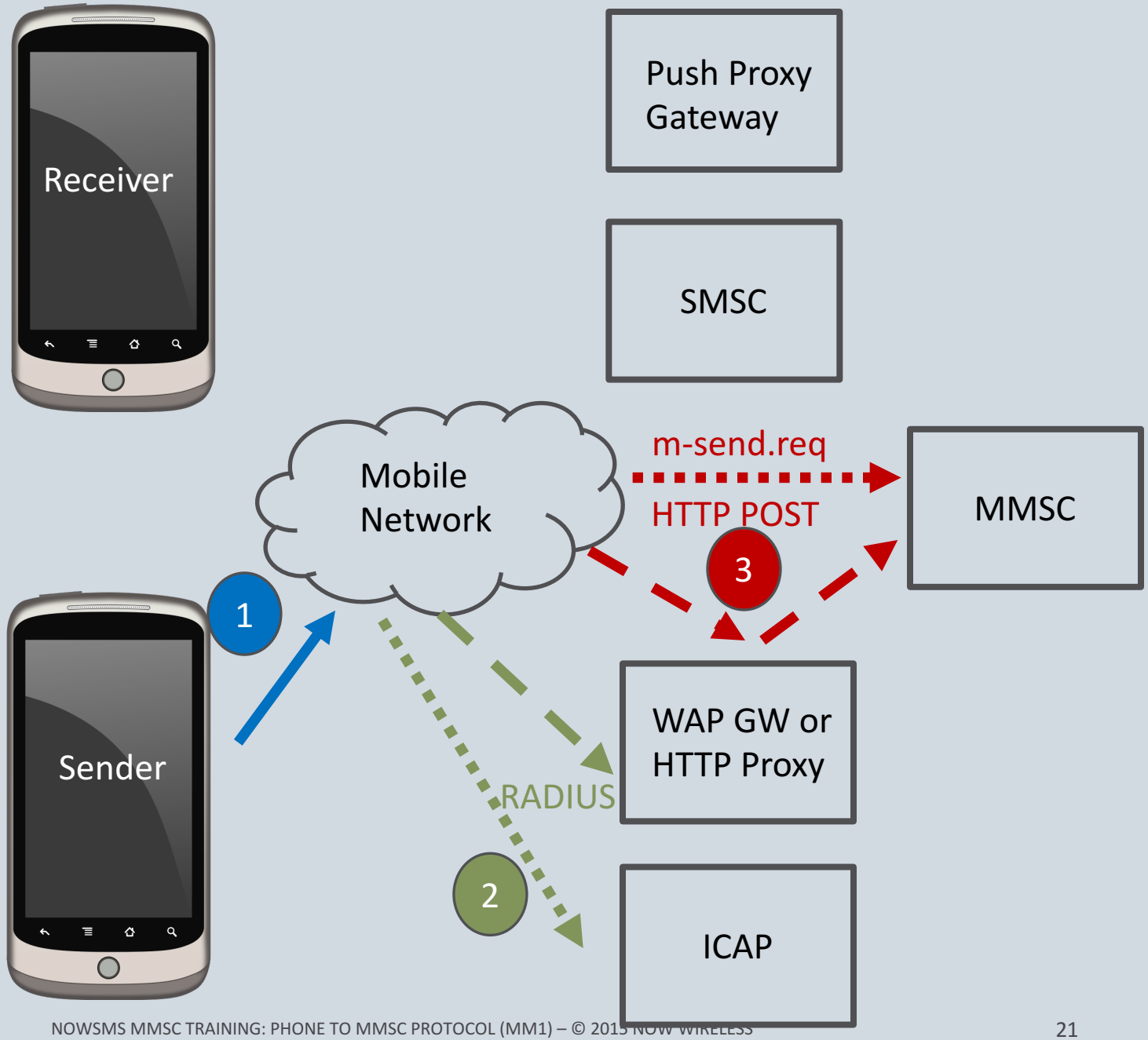
MM1 Transaction: Step 3

Sending phone submits an MM1 m-send.req to the MMSC. (MMSC Server URL configured in phone.)

MMSC receives m-send.req in HTTP POST

Phone may submit:

- WAP1/WSP POST via WAPGW
- WAP2/HTTP POST via Proxy
- HTTP POST direct to MMSC



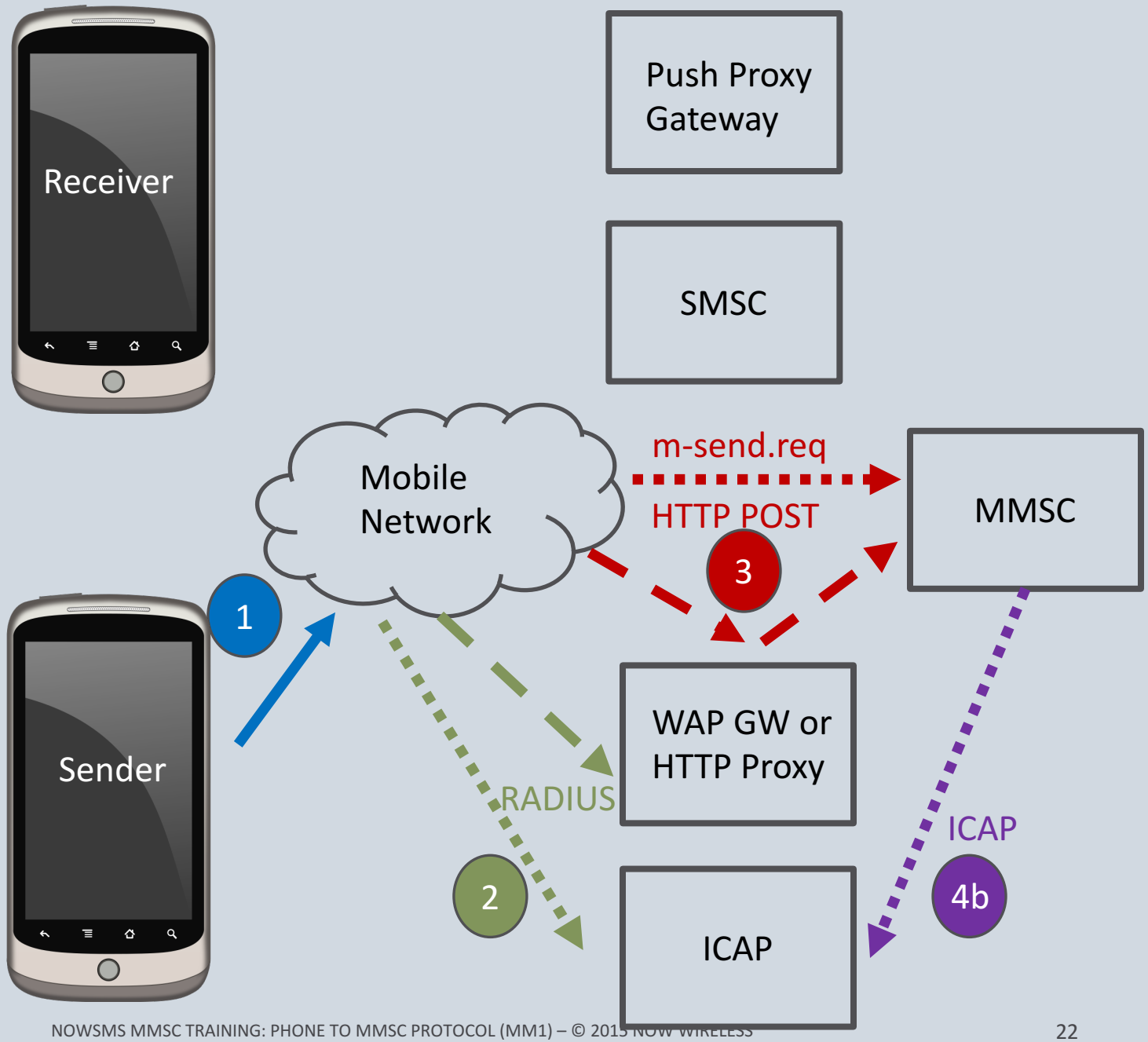
MM1 Transaction: Step 4

MMSC must identify and authenticate sender

MM1 does not define how this is done

Two options:

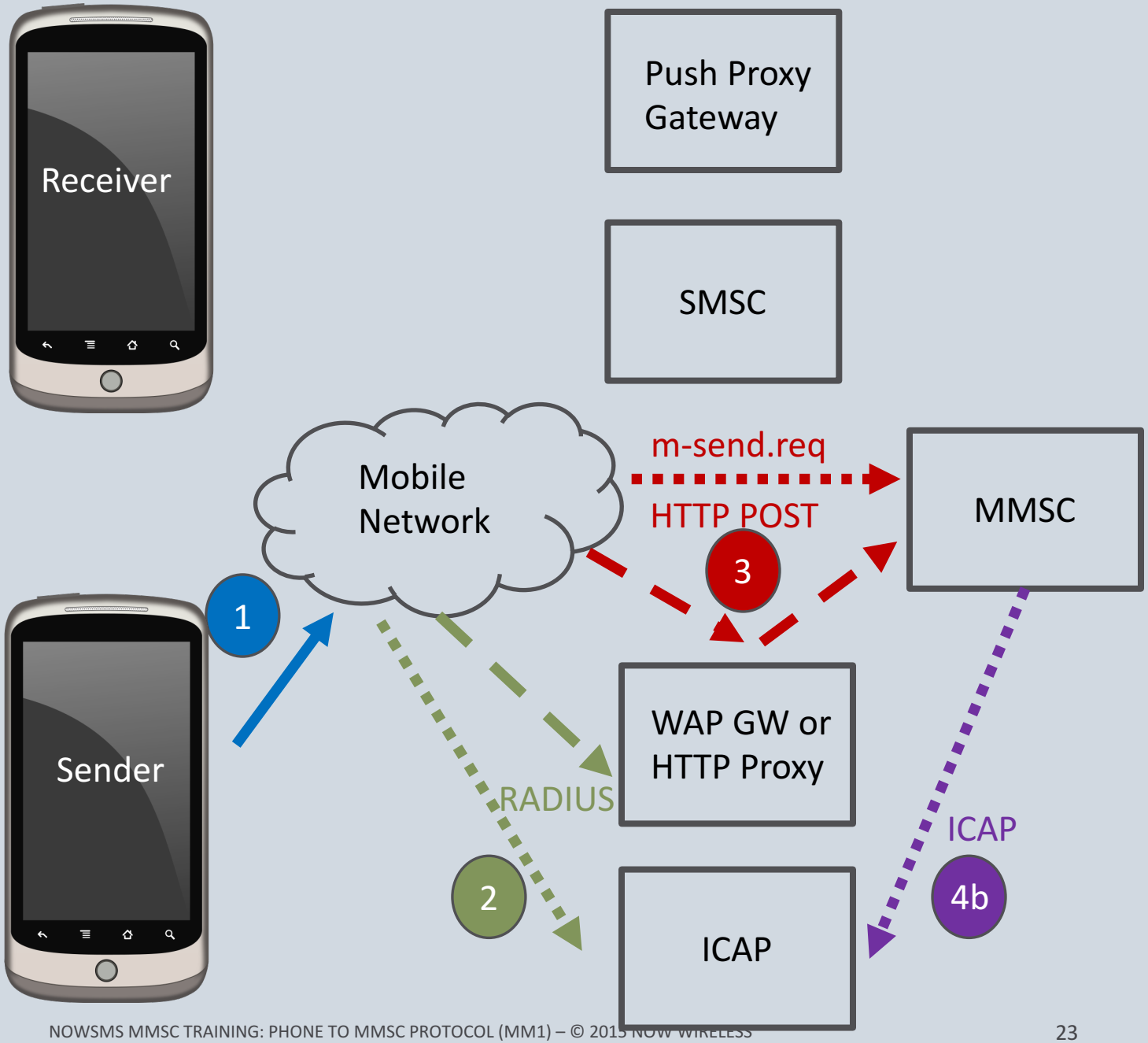
- WAPGW or HTTP Proxy inserts X-MSISDN HTTP header
- MMSC requests MSISDN using ICAP



MM1 Transaction: Step 5

The MMSC accepts and processes the content of the MMS message. If to a local recipient, the MMSC stores and makes it available as a dynamically generated URL link.

The MMSC generates an MMS notification message (m-notification.ind), which is sent via WAP Push over SMS to the recipient(s). This MMS notification message contains a URL pointer to the dynamically generated MMS content.

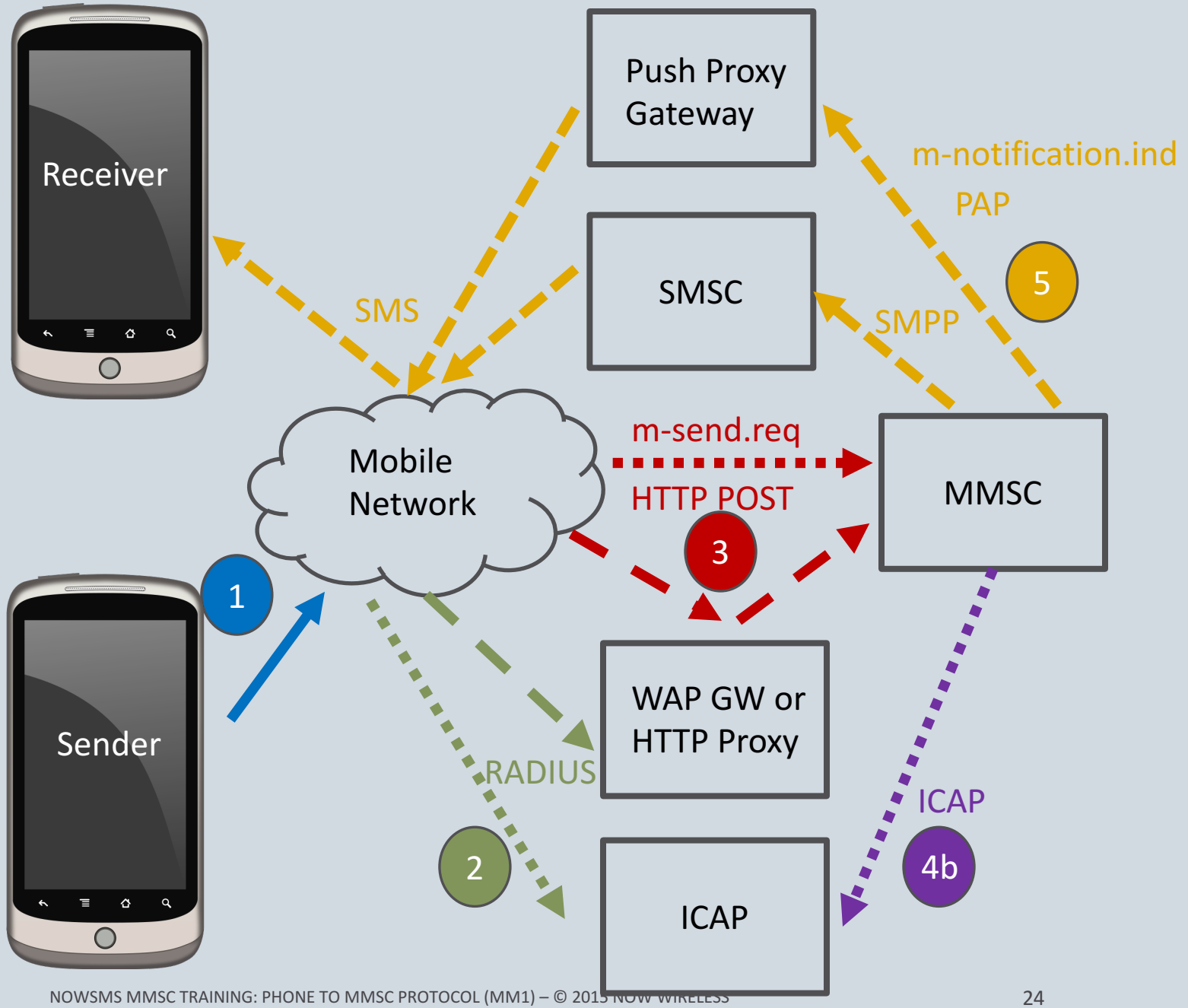


MM1 Transaction: Step 5 (continued)

m-notification.ind WAP push may be submitted to a Push Proxy Gateway using the Push Access Protocol (PAP)

Or it may be encoded directly to SMS format and submitted using SMPP

SMS message generated is usually a 2 part long message

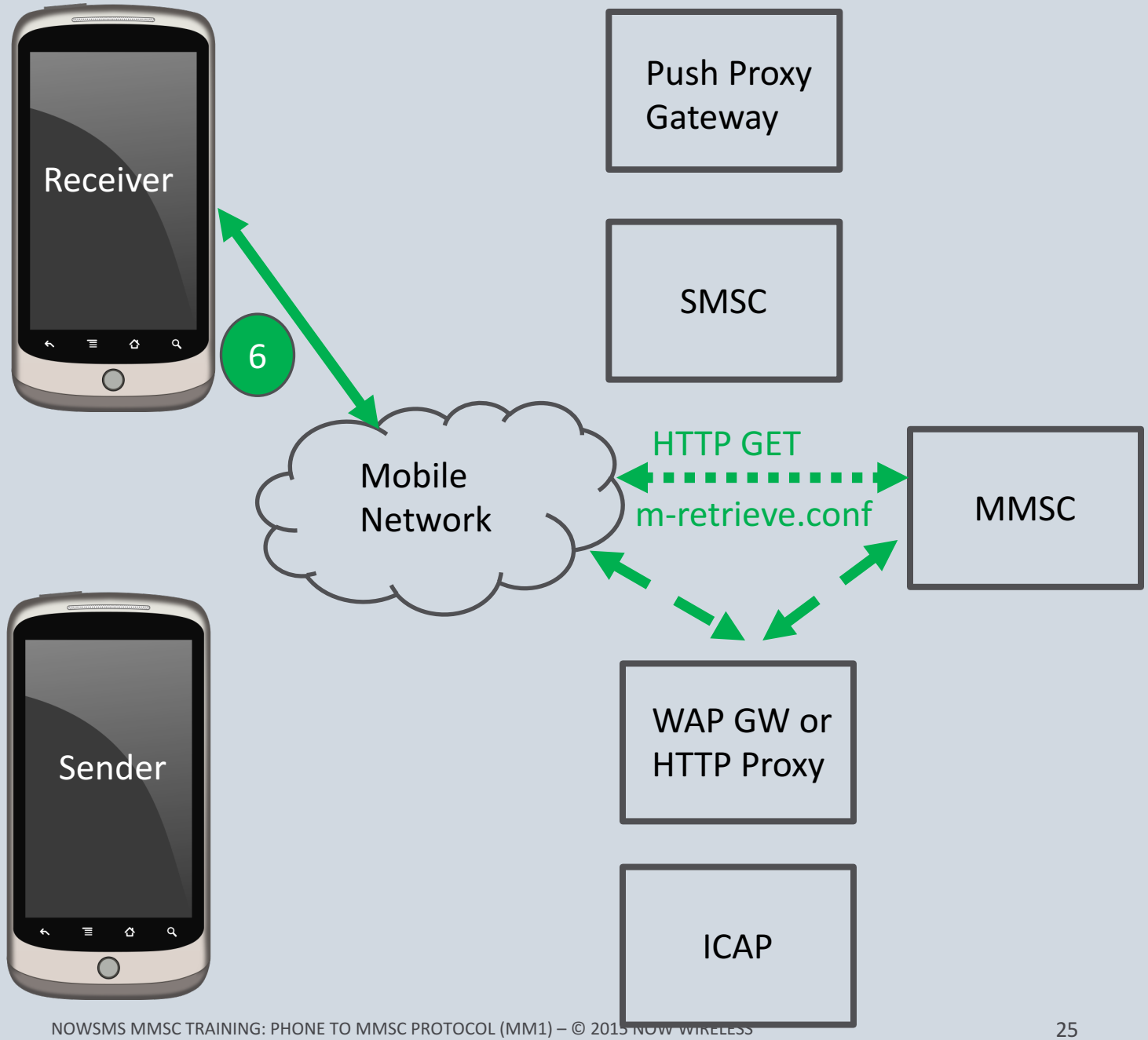


MM1 Transaction: Step 6

The recipient phone activates data connection to MMS APN. (APN setting configured in the phone)

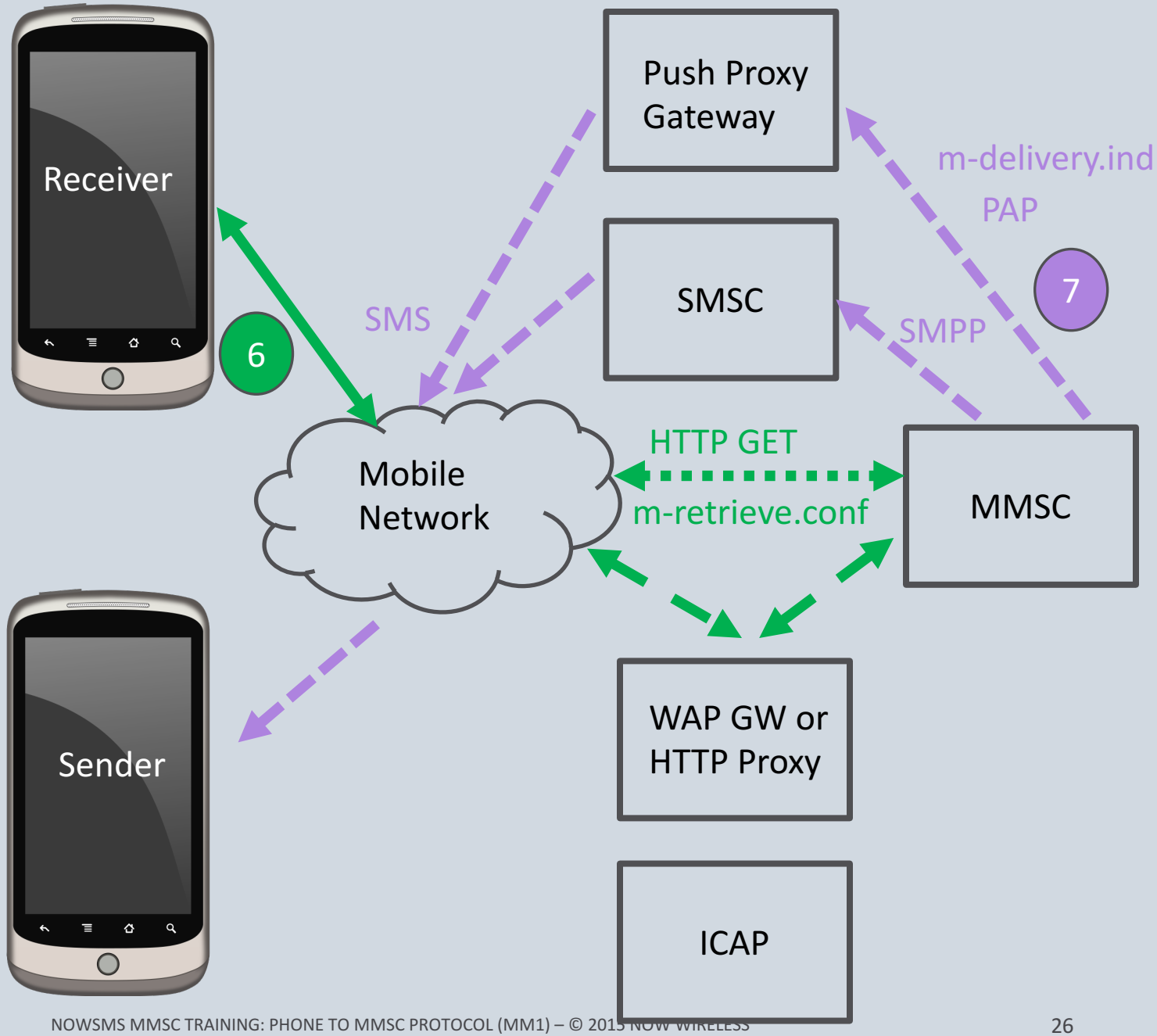
The recipient phone performs an HTTP (or WSP) GET to retrieve the MMS message content URL from the MMSC.

The HTTP response is the MMS message in an m-retrieve.conf PDU.



MM1 Transaction: Step 7

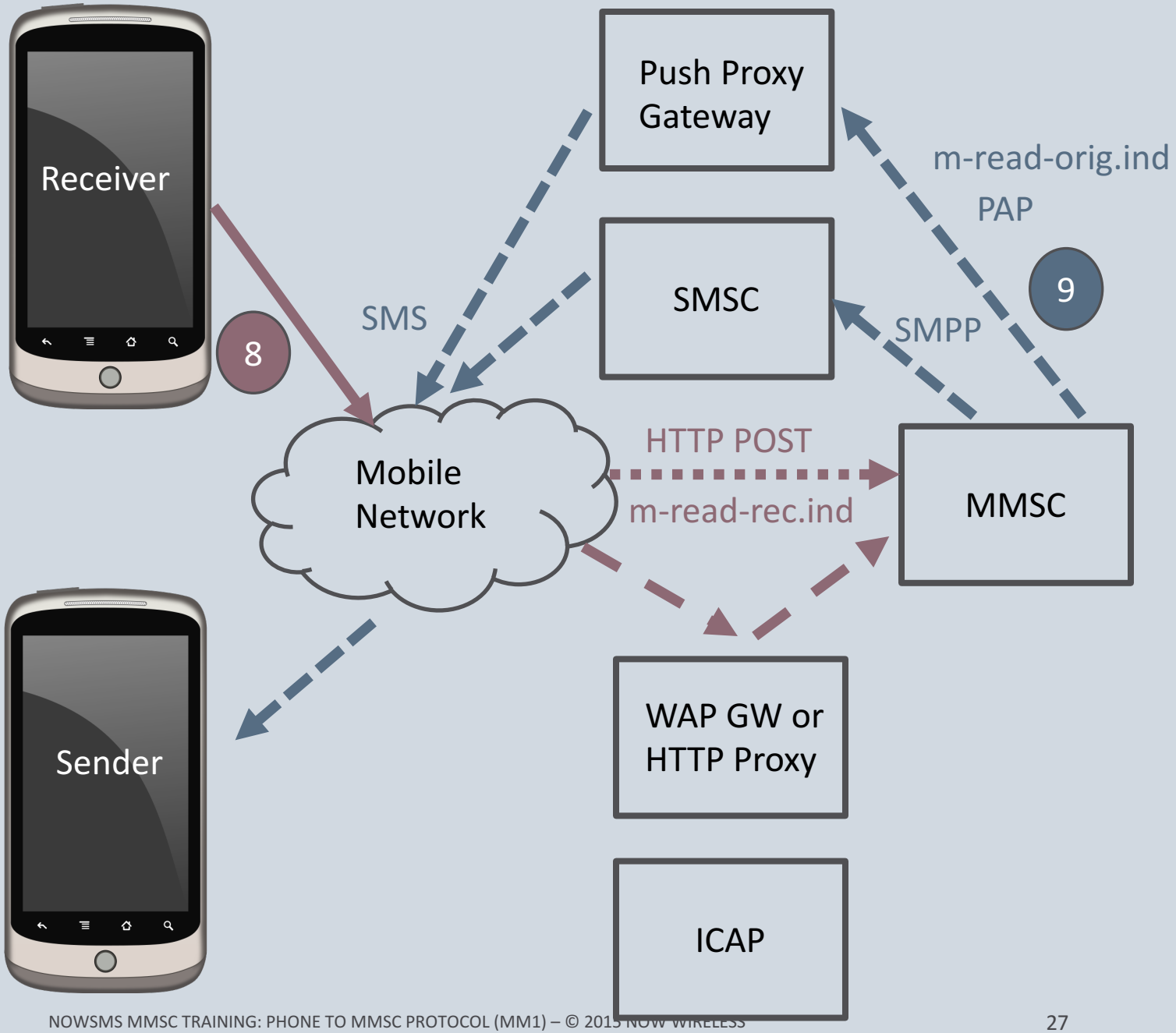
If the sender requested a delivery report, the MMSC generates an MMS delivery report (m-delivery.ind), which is sent via WAP Push over SMS to the original sender.



MM1 Transaction: Step 8

If the sender requested a read report, the receiver may generate a report (m-read-rec.ind) when the subscriber reads the message. (This is dependent on client configuration.)

The MMSC translates the format of the read report (m-read-orig.ind), which is sent via WAP Push over SMS to the original sender.



MMS MO (Mobile-Originated) Flow

1. Sender activates data connection to MMS APN. (APN setting configured in the phone)
2. GGSN sends RADIUS accounting message to notify the network about the device IP assignment.
3. Sending phone submits an MM1 m-send.req to the MMSC. (MMSC Server URL configured in phone.)
4. MMSC must identify and authenticate sender (X-MSISDN header inserted by WAPGW or ICAP)
5. The MMSC accepts and processes the content of the MMS message. If to a local recipient, the MMSC stores and makes it available as a dynamically generated URL link.

MMS MT (Mobile-Terminated) Flow

1. The MMSC generates an MMS notification message (m-notification.ind), which is sent via WAP Push to the recipient(s). This MMS notification message contains a URL pointer to the dynamically generated MMS content.
2. The recipient phone activates data connection to MMS APN. (APN setting configured in the phone)
3. The recipient phone performs an HTTP (or WSP) GET to retrieve the MMS message content URL from the MMSC.
4. Delivery report and/or read report activity may occur

What is WAP Push?

SMS supports a concept of port numbers to allow applications to register to receive messages that are sent to a specific port number.

(WAP Push uses SMS port number 2948)

WAP Push messages are binary messages that can contain different binary content types, one of which is an MMS notification type.

When this MMS notification message is received, the phone routes it to the MMS client for processing.

About WAP Push

WAP Push plays a key role in the MMS delivery process.

By default, the NowSMS MMSC expects to be able to send MMS Notification via its built-in WAP Push Proxy Gateway (PPG), which delivers MMS Notifications using WAP Push over SMS.

For GSM/UMTS/WCDMA environments, NowSMS automatically generates UDH (user data header) in the SMS messages.

For CDMA/CDMA2000 environments, the NowSMS SMPP implementation supports WDP Adaptation so that MMS notification messages can be delivered via SMS using the WAP teleservice.

For other environments, NowSMS can use the Push Access Protocol (PAP) to send MMS notification messages via a separate WAP Push Proxy Gateway (PPG).

MM1 User Authentication Issues

The MM1 Protocol does not define how the MMSC identifies and authenticates the subscriber when the MMS client on a mobile device sends or receives a message.

The MMSC is expected to interface with other operator network components to identify and authenticate the subscriber.

HTTP Header Enrichment

MMS clients send HTTP POST requests to the MMSC over TCP/IP.

Accepted industry practice is to use HTTP Header Enrichment services to insert additional HTTP headers into these requests to provide user identification and authentication.

For example, the subscriber MSISDN, is frequently inserted into one of the following headers:

- `X-MSISDN:`
- `msisdn:`
- `X-MDN:`
- `X-Device-MIN:`

Who performs HTTP Header Enrichment?

The first IP-based mobile operator services were based on WAP technologies, where a WAP Gateway acted as a proxy for all IP-based services.

These WAP gateways often performed HTTP Header Enrichment to allow IP-based services to identify subscribers.

The usual process is to configure the GGSN to send RADIUS Accounting messages to the WAPGW every time a subscriber connects to or disconnects from the GGSN. The WAPGW maintains a table of active connections that allow it to map IP addresses to device phone number for HTTP Header Enrichment.

HTTP Header Enrichment without a GW

As mobile usage has grown, gateways and proxies have become network bottlenecks.

The ICAP protocol allows an HTTP based service to request HTTP Header Enrichment on demand, without forcing clients to interface through a gateway or proxy.

The NowSMS MMSC can be configured to request HTTP HHE via ICAP, and NowWAP can be configured to provide this ICAP based service.