METHODS AND APPARATUS FOR FRAGMENTING SYSTEM INFORMATION MESSAGES IN WIRELESS NETWORKS

BACKGROUND

Field

[0001] The present application relates generally to media delivery in a data network, and to methods and apparatus for fragmenting system information messages for delivery over a wireless network.

Background

[0002] In a content delivery/media distribution system, programming information that describes content and delivery schedule of available content and/or services may be provided to devices in a distribution network. For example, a content distribution network that operates on the media distribution network may provide the programming and/or system information messages to devices in communication with the network. Devices receiving the information operate to display the information to device users who may then subscribe and/or select content and/or services to be received. For example, a device user views the programming guide and/or system information, and may then select and subscribe to receive content and/or services that include multimedia content, clips, programs, scripts, data, customer services, or any other type of content or service.

[0003] Therefore, what is needed is a system that operates to allow large system information messages to be efficiently delivered to devices that may have memory limitation or delivery quality requirements.

SUMMARY

[0004] Methods and apparatus for sending system information (SI) associated with media directed to a device are disclosed. In one aspect, the method includes the steps of fragmenting system information into a plurality of fragments, and transporting the fragments to a device. In another aspect, a method for receiving system information associated with media directed to a device includes receiving system information fragments, and reassembling the fragments to recover the system information.

BRIEF DESCRIPTION OF THE DRAWINGS

[0005] **FIG. 1** shows one embodiment of a system for delivering system information (SI) messages;

[0006] **FIG. 2** shows one embodiment of an SI message fragmentation;

[0007] **FIG. 3** shows one embodiment of a network server for delivering SI messages;

[0008] **FIG. 4** shows one embodiment of a method for delivering SI messages;

[0009] **FIG. 5** shows one embodiment of a device for receiving SI messages; and

[0010] **FIG. 6** shows one embodiment of a method for operating a device for receiving SI messages.

DETAILED DESCRIPTION

[0011] **System information (SI) Distribution Protocols**

[0012] In one embodiment, one or more protocols may be used for the delivery and update of the System Information (SI) on a device. FIG. 1 shows one embodiment for the SI delivery protocols. In one embodiment, there may be two SI delivery protocols:

. The "Marketplace Delivery Protocol" **102**, which may be used to deliver Marketplace and System information (MSI); and

. The "MPG Delivery Protocol" **104**, which may be used to deliver MPG blocks.

[0013] The above protocols may utilize the services of a shared "SI Framing Protocol" **108**.

[0014] The signaling related to a current version of SI messages may be delivered in a "Primary Flow" **106**. The SI framing protocol **108** may utilize the services provided by "Transport Layer protocols" **110**.

[0015] In one embodiment, the upper layer **124** in the network generates, maintains, and updates system information. New or modified SI elements or attributes may be made available to the marketplace and MPG delivery protocol entities for incorporation in SI messages. An SI message is created, or a new or modified element is incorporated in an existing SI message, and the SI version number is updated.

[0016] The marketplace and MPG delivery protocol entities in the network may communicate the latest SI versioning information to the primary flow protocol, and may

schedule the delivery of copies of the SI message over the appropriate multicast or unicast channels, e.g., SI flows.

[0017] The SI framing protocol entity **108** in the network receives SI messages from the marketplace and/or MPG delivery protocol entities. The SI messages may be encoded in XML, SGML, or any other structural markup language text formats.

[0018] In one embodiment, the SI framing protocol entity **108** fragments an SI message into SI message fragments, **112**, if the size of the SI message exceeds a configurable maximum size. The SI message instances or SI message fragments may then be encoded, e.g., to a binary format, **114**, as binary SI messages. The binary SI messages may be subject to additional fragmentation into smaller binary fragments, **116**, for transport. The SI message fragments, the SI messages, the binary fragments, or any combination thereof, are then passed to the transport layer protocol entity **110**, for delivery over a multicast interface, for example.

[0019] The MPG and marketplace delivery protocol entities at the device may receive SI versioning information from the primary flow protocol entity upon activation, and optionally periodically thereafter. When a change to the current version of an SI message is detected, the device may select the corresponding SI flow to acquire the latest information.

[0020] In one embodiment, the SI message fragments, the binary-encoded SI message fragments, the binary fragments, or any combination thereof are received at the device. The received fragments may be binary reassembled **118**, binary decoded **120**, and XML reassembled, **122**. If an SI message instance is received in fragments, the SI framing protocol at the device passes the entire information from the SI message to the marketplace or MPG delivery protocol entity after the constituent SI fragments are received, so that the original SI message may be reassembled.

[0021] **SI Framing Protocol**

- In one embodiment, the SI framing protocol **108** provides four services:
- Fragmentation and reassembly of SI messages, **112** and **122**
- Encoding and decoding of SI messages or fragments, **114** and **120**
- Fragmentation and reassembly of the encoded SI messages or fragments, **116** and **118**; and
- Management of transmission and reception of the encoded SI messages or fragments by the transport layer, **110**.

SI Message Fragmentation

[0022] SI message fragmentation refers to the fragmentation and reassembly of SI Messages. SI fragmentation may be performed to:

. Mitigate the effect of packet loss, and/or

. Accommodate physical, e.g., memory, limitations on the device by allowing the entire received SI message fragment to be loaded in the available memory on the device. In one embodiment, the SI message is made available to the upper layer **126** in the device only after all fragments are received.

[0023] In one embodiment, the network may fragment an SI Message into two or more SI message fragments, if the size of the SI message exceeds a predetermined "SI_Message_Max_Size" parameter. The SI_Message_Max_Size is a configurable network parameter whose value may depend on the transport reliability requirements and the physical, e.g., memory, display size, processor type, etc., limitations on the device. The SI_Message_Max_Size may depend on the transmission technology, i.e., it may be a uniform parameter that accommodates all the device limitations (e.g., memory size) in broadcast transmission, but may vary from device to device in unicast transmission. The maximum allowable loss probability for SI messages may limit the maximum size of SI messages and, therefore; the maximum size of the SI message payload prior to binary encoding. Further, the decoding process and the maximum size of the decoding buffer on the device may also impose a limit on the value of SI_Message_Max_Size.

SI Message Fragment Structure

[0024] In one embodiment, an SI message fragment includes the root attributes of the parent SI message, additional fragment attributes, and one or more atomic elements. An atomic element is an element or sub-element of the message fragment that may not be further fragmented. An SI message fragment may not exceed SI_Message_Max_Size. The number of fragments may not exceed "SI_Fragments_Max_Number," a configurable network parameter whose value depends on the transport reliability requirements and the memory limitations on the device. If it is not possible to fragment an SI message because of either or both of these restrictions, the network may abort the transmission of the SI message.

[0025] The SI message fragment attribute may include a fragment ID and/or the total number of fragments of the parent SI message. An example of the fragmentation

of a "Marketplace Content Retailer" message instance being partitioned into two SI message fragments is depicted in **FIG. 2.** **FIG. 2** shows a parent SI message **202**, and two exemplary SI message fragments **204** and **206** of the parent SI message **202**. The SI message **202** has message root attributes **208** and message atomic elements **210**. The message root attributes **208** may include an SI message ID, an SI message version number, and/or one or more SI message specific fields or keys. The SI message fragment **204** has fragment root attributes **212**, and fragment atomic elements **214**. The fragment root attributes **212** includes its parent message root attributes **208**, e.g., SI message ID, the message version number, and/or one or more message specific fields or keys, and the fragment attributes, e.g., fragment ID (e.g., 1) and the total number of fragments (e.g., 2) of the parent SI message. The SI message fragment **206** has fragment root attributes **216**, and fragment atomic elements **218**. The fragment root attributes **216** includes its parent message root attributes **208**, e.g., SI message ID, the message version number, and/or one or more message specific fields or keys, and the fragment attributes, e.g., fragment ID (e.g., 2) and the total number of fragments (e.g., 2) of the parent SI message.

[0026] The atomic elements of an SI message instance depend on the type of SI message, among other possible parameters. Each direct sub-element of an SI message instance is an atomic element. Table 1 lists the atomic elements of the listed SI message types.

Table 1 Atomic Elements

SI Message	Atomic Elements
Marketplace Common	Classification Scheme Table BCS Record
Marketplace Content Retailer	Basic Info EULA Table Package Record Tier Record Channel Record Auxiliary Data
Service Definition	Service Record Auxiliary Service Record
MPG Block	MPG Title Record Channel Customization Record Contact Window Blackout Record

Fragment Attributes

[0027] In one embodiment, two root attributes are defined for an SI message Fragment:

- Fragment ID
- Number of fragments

[0028] These attributes may be present in the SI message fragments, but they are not present in un-fragmented SI messages.

[0029] The fragment ID attribute uniquely distinguishes the SI message fragment from all other SI message fragments of the same version of an SI message. The fragment ID attribute may be an 8-bit unsigned integer, for example. The value of the fragment ID attribute may be set to "1" for the first SI message fragment, and may be incremented, e.g., by 1, for each subsequent fragment of the same SI message instance. The value of the fragment ID may not exceed SI_Fragments_Max_Number.

[0030] The number-of-fragments attribute specifies the number of SI message fragments of an SI message instance. The number-of-fragments attribute may be an 8-bit unsigned integer, for example. The value of the number-of-fragments attribute may be equal to the maximum value of the fragment ID attribute used by the SI message fragments of the version of the SI message being fragmented. The minimum value of the number-of-fragments attribute is 2. The number-of-fragments attribute has the same value in all SI message fragments of the same version of the SI message.

Encoding of SI Message Fragments

[0031] Each SI message or SI message fragment may be encoded to a second language representation, e.g., binary, as shown in **FIG. 1, 114**. The binary encoding algorithm may include "ASN.1 Basic PER" algorithm, as specified in ISO/IEC 8825-2. Both aligned and unaligned options may be supported.

Fragmenting Encoded SI Message Fragments

[0032] The network may divide each encoded SI message into one or more (e.g., binary) fragments. The binary fragments except the last one may be of the same size. The size of a binary fragment may be specified by a system parameter Binary_SI_Message_Fragment_Size. For example, a binary SI message may be fragmented into 255 binary fragments. Each binary fragment may be prefaced by a header that allows the device to identify each fragment and reassemble the original binary SI message. The device reassembles, **118**, each binary SI message before

decoding it. One embodiment of the format of the binary fragment header is shown in Table 2.

[0033] Table 2 Binary SI Fragment Header Format

Field Name	Field Type	
MESSAGE_ID	UINT(n)	
MESSAGE_SPECIFIC_FIELDS	VARIABLE	
FRAGMENT_ID	UINT(n)	
TOTAL_FRAGMENTS	UINT(n)	

[0034] Where, UINT stands for Unsigned Integer (n bits). The fields of the binary SI fragment header are defined in the following subsections.

MESSAGE_ID

[0035] This field identifies the type of SI message being fragmented. Some values for the MESSAGE_ID field are described in Table 3.

Table 3 Binary SI fragment MESSAGE_ID values

SI Message Type	MESSAGE_ID Value
SERVICE_DEFINITION	1
MARKETPLACE_COMMON	2
CONTENT_RETAILER_MARKETPLACE	3
MPG_BLOCK	4

[0036] For example, the MESSAGE_ID field is set to SERVICE_DEFINITION if the SI message being fragmented is a service definition SI message.

MESSAGE_SPECIFIC_FIELDS

[0037] The MESSAGE_SPECIFIC_FIELDS or keys convey the values of the SI message fields that distinguish different SI messages. The set of fields involved is specific to each type of SI message. Accordingly, the size of the MESSAGE_SPECIFIC_FIELDS varies from 2 bytes to 5 bytes according to the value of MESSAGE_ID. In one embodiment, a format of the MESSAGE_SPECIFIC_FIELDS when the MESSAGE_ID is set to MARKETPLACE_COMMON or SERVICE_DEFINITION is shown in Table 4.

Table 4 Binary SI fragment MESSAGE_SPECIFIC_FIELDS -- MARKETPLACE_COMMON and SERVICE_DEFINITION messages

Field Name	Field Type	
VERSION	UINT(n)	

[0038] In one embodiment, a format of the MESSAGE_SPECIFIC_FIELDS when the MESSAGE_ID is set to CONTENT_RETAILER_MARKETPLACE is shown in Table 5.

Table 5 Binary SI fragment MESSAGE_SPECIFIC_FIELDS -- CONTENT_RETAILER_MARKETPLACE messages

Field Name	Field Type	
CONTENT_RETAILER_ID	UINT(n)	
VERSION	UINT(n)	

[0039] In one embodiment, a format of the MESSAGE_SPECIFIC_FIELDS when the MESSAGE_ID is set to MPG_BLOCK is shown in Table 6.

Table 6 Binary SI fragment MESSAGE_SPECIFIC_FIELDS -- MPG_BLOCK messages

Field Name	Field Type	
MPG_BLOCK_START_TIME	UINT(n)	
MPG_BLOCK_VERSION	UINT(n)	

FRAGMENT_ID

[0040] Each fragment of a message is identified by the FRAGMENT_ID. This field may be used by the device to locate the position of the fragment in the binary SI message and to determine when it has received all the required fragments of the message.

[0041] Fragments may be numbered sequentially according to their position in the binary SI message, e, g., starting with 0. The value of the last fragment would be equal to TOTAL_FRAGMENTS – 1. For example, when a binary SI message is fragmented into 255 fragments, the value of FRAGMENT_ID may not exceed 254.

TOTAL_FRAGMENTS

[0042] This field indicates the total number of fragments of an SI message. For example, the range of values for this field is 1 through 255.

Distribution Algorithm

[0043] The network may transmit the binary SI message fragments of a given version of an SI message at least once before starting transmission of the next SI message on the same SI flow. The maximum interval between consecutive message

fragment transmissions may not exceed $T_{FRAGMENT_ACQUISITION}$ (ms) parameter. $T_{FRAGMENT_ACQUISITION}$ is a configurable system parameter.

Acquisition of SI Message Fragments

[0044] A device which acquires an SI message in which the fragment attributes are present determines that the SI message is an SI Message fragment. The device may acquire the SI message fragments of a version of an SI message before processing the entire SI message.

Marketplace Delivery Protocol

[0045] The marketplace delivery protocol may deliver and/or update messages, such as the following SI messages:

- Marketplace Common Message
- Marketplace Content Retailer Message (per Content Retailer)
- Service Definition Message

[0046] The above SI messages are collectively referred to as Marketplace & System information (MSI). The network may deliver MSI corresponding to the Wide-area Operations Infrastructure (WOI), and if available, to Local-area Operations Infrastructure (LOI) multiplexes. The MSI pertaining to a WOI or LOI multiplex may be delivered over the corresponding WOI or LOI marketplace definition SI flows. The network may signal, e.g., on the primary flow, presence of the MSI on the marketplace definition SI flows, and the current Version of the MSI on each marketplace definition SI flow. The MSI may be transmitted cyclically, in a predetermined order. The maximum interval between consecutive transmissions of MSI messages may not exceed $T_{MARKETPLACE_ACQUISITION}$ (ms). The $T_{MARKETPLACE_ACQUISITION}$ may be a configurable system parameter. A device may acquire the MSI delivered on the WOI marketplace definition flow or on the LOI marketplace definition flow, if any present. The device may determine the current version of the MSI from the primary flow, and may detect an update to any MSI message as a change of version for that message in the primary flow.

Media Presentation Guide Delivery Protocol

[0047] The Media Presentation Guide (MPG) may provide a user with a schedule of what will be available for viewing on each Service. If the MPG Information is tied to a given time period, the network continuously delivers and updates the device with the latest MPG. The network may deliver MPG blocks for MPG titles transmitted in the WOI and, if available, in the LOI multiplexes. The MPG titles transmitted in a WOI or

LOI multiplex may be delivered over the corresponding WOI or LOI Near-term and/or Far-term MPG SI flows. MPG block messages on each MPG flow may be transmitted cyclically, e.g., in ascending order of the value of the "Start_Time" attribute of the MPG block. The MPG block message may specify the "MPG_Block_Start_Time," which is the earliest time covered by the MPG block. The MPG_Block_Start_Time of each MPG blocks corresponds to the end of the interval covered by the previous MPG lock.

[0048] The maximum interval between consecutive transmissions of MPG block messages may not exceed $T_{MPG_ACQUISITION}$ (ms). The $T_{MPG_ACQUISITION}$ may be a configurable system parameter. The network may stop transmission of an MPG block when the "System Time" exceeds the "Start_Time" of the MPG Block by more than "MPG_Block_Duration." The MPG title record may specify the MPG_Block_Duration. If the service is a real-time service or an IP-datacast service, the MPG_Block_Duration added to the MP_ Block_Start _Time is the time at which display of the content may end. If the service is a non-real-time service, the MPG_Block_Duration added to the MPG_Block_Start_Time is the latest time at which display of the content may commence, exclusive of any introductions associated with the MPG title. If the service is a "Per MPG Title" service, the significance of the MPG_Block_Start_Time is dependent on the nature of the content associated with the MPG title, as defined in the preceding two paragraphs.

[0049] The near-term MPG SI flow may be used to transmit the nearest MPG blocks applicable to a multiplex. The number of MPG blocks in the near-term MPG SI flow may not be less than MPG_Min_Num_Multicast_Blocks, where MPG_Min_Num_Multicast_Blocks is a configurable network parameter. The far-term MPG SI flow is used to transmit MPG blocks applicable to the multiplexes that are not transmitted in the near term MPG SI flow. The total number of MPG blocks in the near and far-term MPG SI flows combined may not exceed MPG_Max_Num_Multicast_Blocks, where MPG_Max_Num_Multicast_Blocks is a configurable network parameter.

MPG Block Version Management

[0050] The network may maintain a MPG_Version parameter, which may be incremented whenever:

- An MPG block is added to the near-term or far-term MPG SI flow,

. An MPG block is removed from the near-term or far-term MPG SI flow, and/or

. The version of any MPG block is changed,

[0051] The network may signal the current value of the MPG_Version to the device through the primary flow, to signal a change to at least one MPG block, the addition or deletion of an MPG block, or the transfer of an MPG block from a far-erm MPG SI flow to a near-term MPG SI flow.

MPG Distribution in Primary Flow

[0052] The network may signal the current values of the following parameters through the primary flow:

. The presence or absence of an MPG SI flow,

. MPG_Block_Duration,

. The Start_Time of the earliest MPG block message currently being transmitted,

. The number of MPG blocks currently being transmitted in the near–term flow,

. The total number of MPG blocks currently being transmitted in the near–term and far-term flows,

. The MPG_Version, and

. The versions of each MPG block currently being transmitted.

[0053] The device may use these parameters to control initial acquisition of the MPG blocks, to detect the expiration, addition, deletion or change of MPG blocks, and to acquire updated versions of MPG blocks. The device may acquire and store at least the nearest MPG_Min_Num_Stored_Blocks MPG Blocks. The device may determine the current version of the MPG blocks and the availability of new MPG blocks from the primary flow.

[0054] **FIG. 3** shows one embodiment of a network server **300** for use in one embodiment of a delivery system for delivering SI messages. The server **300** comprises processing logic **302** and transceiver logic **304**, which are coupled to an internal data bus **306**. The server **300** also comprises encoder logic **308** and fragments generation logic **310**.

[0055] In one or more embodiments, the processing logic **302** comprises a CPU, processor, gate array, hardware logic, memory elements, virtual machine, software, and/or any combination of hardware and software. Thus, the processing logic **302** generally comprises logic to execute machine-readable instructions and to control one or more other functional elements of the server **300** via the internal data bus **306**.

[0056] The transceiver logic **304** comprises hardware logic and/or software that operate to allow the server **300** to transmit and receive data and/or other information with remote devices or systems using communication channel **312**. For example, in one embodiment, the communication channel **312** comprises any suitable type of communication link to allow the server **300** to communicate with one or more data networks. For example, in one embodiment, the transceiver logic **304** operates to receive SI messages from one or more remote content servers or protocols. The server **300** then operates to fragment and or encode the SI messages that are transmitted to devices operating on one or more wide area networks.

[0057] Therefore, the server **300** operates in one or more embodiments of a delivery system to deliver SI messages to devices operating on one or more wide area networks. It should be noted that the server **300** illustrates just one implementation and that other implementations are possible within the scope of the embodiments.

[0058] **FIG. 4** shows one embodiment of a method **400** for operating a network server in one embodiment of a SI-message delivery system. For clarity, the method **400** will be described with reference to the network server **300** shown in **FIG. 3** and **FIG. 1**. In one embodiment, at least one processor, such as the processing logic **302**, executes machine-readable instructions to control the server **300** to perform the functions described below. At block **402**, one or more SI messages are received for transmission to one or more devices. The SI messages may be expressed in a first language representation, e.g., XML. For example, one or more content providers provide one or more SI messages for distribution to one or more devices. In one embodiment, the SI messages are received from the marketplace delivery protocol **102** and/or from MPG delivery protocol **104**. At block **404**, one or more SI messages are fragmented within the first representation. At block **406**, one or more SI fragments may be encoded from the first representation to a second representation, e.g., binary. At block **408**, one or more of the encoded fragments may be further fragmented within the second representation. At block **410**, the fragments are transmitted to one or more devices.

Thus, the method **400** operates to deliver SI messages to one or more devices with memory-size limitations. It should be noted that the method **400** represents just one implementation and that other implementations are possible within the scope of the embodiments.

[0059] **FIG. 5** shows one embodiment of a device **500** for use in one embodiment of a system for delivering SI messages. The device **500** comprises processing logic **502**, device resources and interface logic **504**, and transceiver logic **506**, which are coupled to an internal data bus **508**. The device **500** also comprises decoding logic **510** and reassembly logic **512**, which are also coupled to the data bus **508**. In one or more embodiments, the processing logic **502** comprises a CPU, processor, gate array, hardware logic, memory elements, virtual machine, software, and/or any combination of hardware and software. Thus, the processing logic **502** generally comprises logic to execute machine-readable instructions and to control one or more other functional elements of the device **500** via the internal data bus **508**.

[0060] The device resources and interfaces logic **504** comprise hardware and/or software that allow the device **500** to communicate with internal and external systems. For example, the internal systems may include mass storage systems, memory, display driver, modem, or other internal device resources. The external systems may include user interface devices, displays, printers, disk drives, keyboard, keypad, cursor keys, pointing device, or any other local devices or systems. For example, the device interface logic **504** operates to receive user inputs from a keypad, and output information to be displayed on a device display.

[0061] The transceiver logic **506** comprises hardware logic and/or software that operate to allow the device **500** to transmit and receive data and/or other information with remote devices or systems using communication channel **514**. For example, in one embodiment, the communication channel **514** comprises any suitable type of communication link to allow the device **500** to communicate with one or more data networks. For example, in one embodiment, the transceiver logic **506** operates to receive SI messages and/or fragments from one or more remote servers. The SI messages and/or fragments received may then be processed by decoding logic **510** and/or reassembly logic **512**.

[0062] In one embodiment, the delivery system comprises program instructions stored on a computer-readable medium, which when executed by at least one processor,

for instance, the processing logic **502**, provides the functions described herein. For example, the program instructions may be loaded into the device **500** from a computer-readable media, such as a floppy disk, CDROM, memory card, FLASH memory device, RAM, ROM, or any other type of memory device or computer-readable medium that interfaces to the device **500**. In another embodiment, the instructions may be downloaded into the device **500** from an external device or network resource that interfaces to the device **500** through the transceiver logic **506**. The program instructions, when executed by the processing logic **502**, provide one or more embodiments of a delivery system.

[0063] Therefore, the device **500** operates in one or more embodiments of a delivery system to receive SI messages and/or fragments from a network server. It should be noted that the device **500** illustrates just one implementation and that other implementations are possible within the scope of the embodiments.

[0064] **FIG. 6** shows one embodiment of a method **600** for operating a device in one embodiment. For clarity, the method **600** will be described with reference to the device **500** shown in **FIG. 5**, and **FIG.1**. In one embodiment, at least one processor, such as the processing logic **502**, executes machine readable instructions to control the device **500** to perform the functions described below.

[0065] At block **602**, message fragments are received, which may have been through fragmentation in a first (e.g., XML) and/or a second (binary) language representation, at the network, **116**. At block **604**, it is determined whether the received fragments had been fragmented within a second representation. If yes, the received fragments are reassembled within the second representation, in step **606**. At block **608**, it is determined whether the fragments had been encoded from a first representation to the second representation. If yes, the fragments are decoded from the second representation to the first representation, in step **610**. At block **612**, it is determined whether the SI message had been fragmented within the first representation, e.g., XML. If yes, the fragments are reassembled within the first representation, in step **614**. At block **616**, the recovered SI messages are delivered to upper layer, e.g., marketplace delivery protocol and/or MPG delivery protocol.

[0066] Thus, the method **600** operates to allow a device to receive a SI messages in one embodiment of a delivery system. It should be noted that the method **600**

represents just one implementation and that other implementations are possible within the scope of the embodiments.

[0067] The various illustrative logics, logical blocks, modules, and circuits described in connection with the embodiments disclosed herein may be implemented or performed with a general purpose processor, a digital signal processor (DSP), an application specific integrated circuit (ASIC), a field programmable gate array (FPGA) or other programmable logic device, discrete gate or transistor logic, discrete hardware components, or any combination thereof designed to perform the functions described herein. A general-purpose processor may be a microprocessor, but, in the alternative, the processor may be any conventional processor, controller, microcontroller, or state machine. A processor may also be implemented as a combination of computing devices, e.g., a combination of a DSP and a microprocessor, a plurality of microprocessors, one or more microprocessors in conjunction with a DSP core, or any other such configuration.

[0068] The steps of a method or algorithm described in connection with the embodiments disclosed herein may be embodied directly in hardware, in a software module executed by a processor, or in a combination of the two. A software module may reside in RAM memory, flash memory, ROM memory, EPROM memory, EEPROM memory, registers, a hard disk, a removable disk, a CD-ROM, or any other form of storage medium known in the art. An exemplary storage medium is coupled to the processor, such that the processor can read information from, and write information to, the storage medium. In the alternative, the storage medium may be integral to the processor. The processor and the storage medium may reside in an ASIC. The ASIC may reside in a user terminal. In the alternative, the processor and the storage medium may reside as discrete components in a user terminal.

[0069] The description of the disclosed embodiments is provided to enable any person skilled in the art to make or use the present invention. Various modifications to these embodiments may be readily apparent to those skilled in the art, and the generic principles defined herein may be applied to other embodiments, e.g., in an instant messaging service or any general wireless data communication applications, without departing from the spirit or scope of the invention. Thus, the present invention is not intended to be limited to the embodiments shown herein but is to be accorded the widest scope consistent with the principles and novel features disclosed herein. The word

"exemplary" is used exclusively herein to mean "serving as an example, instance, or illustration." Any embodiment described herein as "exemplary" is not necessarily to be construed as preferred or advantageous over other embodiments.

CLAIMS

WHAT IS CLAIMED IS:

1. A method for sending system information messages expressed in a first language representation, associated with media directed to a device, the method comprising:

fragmenting at least one system information (SI) message into a plurality of fragments within the first language representation, wherein each fragment carries at least one root attribute of the SI message; and

transporting the fragments to a device.

2. The method of claim 1, further comprising encoding at least one fragment into a second language representation.

3. The method of claim 2, further comprising fragmenting at least one of the fragments within the second language representation.

4. The method of claim 3, wherein said encoding comprises encoding the fragments from XML representation into binary representation.

5. The method of claim 1, wherein said fragmenting comprises fragmenting the system information if the system information has a larger size than a predetermined limit.

6. The method of claim 5, wherein the predetermined limit is determined based on at least one physical characteristic of the device.

7. The method of claim 6, wherein the physical characteristic comprises available memory on the device.

8. A method for receiving system information associated with media directed to a device, the method comprising:

receiving system information (SI) fragments, the fragments carrying at least one attribute of a parent SI message expressed in a first language representation; and

reassembling the fragments to recover the parent SI message.

9. The method of claim 8, wherein if the fragments comprise fragments expressed in a second language representation, said reassembling comprises reassembling the fragments within the second language representation.

10. The method of claim 9, wherein if the fragments comprise fragments encoded within the second language representation, further comprising decoding the fragments to the first language representation.

11. The method of claim 8, further comprising storing each fragment on the device, wherein each fragment is sized suitably for the memory on the device.

12. An apparatus for sending system information messages expressed in a first language representation, associated with media directed to a device, comprising:

means for fragmenting at least one system information (SI) message into a plurality of fragments within the first language representation, wherein each fragment carries at least one root attribute of the SI message; and

means for transporting the fragments to a device.

13. The apparatus of claim 12, further comprising means for encoding at least one fragments into a second language representation.

14. The apparatus of claim 13, further comprising means for fragmenting at least one of the fragments within the second language representation.

15. The apparatus of claim 14 wherein said means for encoding comprises means for encoding the fragments from XML representation into binary representation.

16. The apparatus of claim 12, wherein said means for fragmenting comprises means for fragmenting the system information if the system information has a larger size than a predetermined limit.

17. The apparatus of claim 16, wherein the predetermined limit is determined based on at least one physical characteristic of the device.

18. The apparatus of claim 17, wherein the physical characteristic comprises available memory on the device.

19. An apparatus for receiving system information associated with media directed to a device, comprising:

means for receiving system information (SI) fragments, the fragments carrying at least one attribute of a parent SI message expressed in a first language representation ; and

means for reassembling the fragments to recover the parent SI message.

20. The apparatus of claim 19, wherein said means for reassembling comprises means for reassembling the fragments within a second language representation if the fragments comprise fragments expressed in the second language representation.

21. The apparatus of claim 20, further comprising means for decoding the fragments to the first language representation if the fragments comprise fragments encoded within the second language representation.

22. A processor configured to perform a method for sending system information messages expressed in a first language representation, associated with media directed to a device, the method comprising:

fragmenting at least one system information (SI) message into a plurality of fragments within the first language representation, wherein each fragment carries at least one root attribute of the SI message; and

transporting the fragments to a device.

23. A processor configured to perform a method for receiving system information associated with media directed to a device, the method comprising:

receiving system information (SI) fragments, the fragments carrying at least one attribute of a parent SI message expressed in a first language representation ; and

reassembling the fragments to recover the parent SI message.

24. A program resident in a machine-readable medium that, when executed, directs a machine to perform a method for sending system information messages expressed in a first language representation, associated with media directed to a device, the method comprising:

fragmenting at least one system information (SI) message into a plurality of fragments within the first language representation, wherein each fragment carries at least one root attribute of the SI message; and

transporting the fragments to a device.

25. A program resident in a machine-readable medium that, when executed, directs a machine to perform a method for receiving system information associated with media directed to a device, the method comprising:

receiving system information (SI) fragments, the fragments carrying at least one attribute of a parent SI message expressed in a first language representation ; and

reassembling the fragments to recover the parent SI message.