

Hindawi Publishing Corporation
International Journal of Digital Multimedia Broadcasting
Volume 2016, Article ID 3154801, 13 pages
<http://dx.doi.org/10.1155/2016/3154801>



Research Article

Model Aspects of Open Access to Multimedia Broadcast Services in the Evolved Packet System

Ivaylo Atanasov and Evelina Pencheva

Technical University of Sofia, 8 Kliment Ohridski Boulevard, 1000 Sofia, Bulgaria

Correspondence should be addressed to Evelina Pencheva; enp@tu-sofia.bg

Received 23 October 2015; Revised 29 December 2015; Accepted 6 January 2016

Academic Editor: Hwang JN

Copyright © 2016 I. Atanasov and E. Pencheva. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Multimedia broadcast is the most efficient method to distribute identical content to multiple users in the Evolved Packet System (EPS). EPS enables efficient usage of network resources and provisioning of quality of service for every user. Third-party control allows applications in an enterprise domain to invoke network functions like multimedia broadcast. In this paper, an approach to modeling the behavior of Service Capability Server (SCS) for multimedia broadcast in EPS is presented. Third-party applications can access multimedia broadcasting capabilities by using Parlay X Web Service interfaces. The SCS for multimedia broadcast exposes Parlay X interfaces toward 3rd-party applications and control protocols toward the network. The SCS functional behavior has to be synchronized with the application view on message broadcast and the state of the network resources intended for the broadcast session. Models of multicast session, IP connectivity session, and bearers' and charging session are proposed and formally described using the notation of Label Transition Systems. The concept of weak bisimilarity is used to prove that models expose equivalent behavior; that is, they are synchronized.

1. Introduction

The Evolved Packet System (EPS) is standardized to provide ubiquitous access to multimedia services from any end user device. It encompasses both access network and core network. The multimedia packet core network plays an important role in providing superior user experience from services and applications. It is in charge of provisioning quality of service (QoS) for multimedia bearer services and charging mechanisms.

Multimedia broadcast/multicast service (MBMS) allows data to be transmitted to multiple endpoints. It allows optimizing network resources using a distribution mechanism. The broadcast mode is a unidirectional point-to-multipoint transmission of multimedia data (e.g., text, audio, picture, and video) from a single source entity to all users in a broadcast service area. A broadcast service received by the end user device involves one or more successive broadcast sessions. A broadcast service might, for example, consist of a single ongoing session (e.g., a media stream) or may involve several

intermittent sessions over an extended period of time [1, 2]. Examples of content appropriate to a distribution scheme are broadcasting TV channels and distribution of area-based multimedia messaging service including commonly interested traffic, weather, or emergency information. The content provider (broadcast/multicast source) may provide discrete and continuous media, as well as service descriptions and control data. The multimedia content may be provided by a 3rd-party content provider.

The broadcast mode is intended to efficiently use EPS resources. Current research on MBMS covers different aspects related to optimisation of radio technology used for data transmission. Relatively low attention is paid on MBMS third-party control, which allows third-party providers from an IT domain to create applications that use network connections, streaming, messaging, and multimedia.

In this paper, we discuss some model aspects of deployment open access on multimedia message broadcast service in EPS. The open service access allows third-party applications to use broadcast function in the network. The Parlay X

Web Services (WS) model provides a high level of abstraction of communication network functions and it facilitates the development of value added applications.

The paper is organized as follows. In Section 2, the related work is presented and the novelty of the proposed approach is highlighted. In Section 3, the broadcast service architecture in EPS is discussed where the Broadcast Multicast Service Centre (BMSC) mediates between third-party applications and the network infrastructure. Next, the Parlay X Message Broadcast WS functionality is analyzed in Section 4. The behavior of BMSC is modeled by formal descriptions of MBMS session state and message broadcast status in Section 5. In Section 6, it is proved that models considering QoS and charging aspects of MBMS are synchronized. The conclusion summarizes the contribution.

2. Related Work

Research community has studied different architectural aspects of MBMS deployment in next generation networks. In [3], the authors propose service architecture for efficient content delivery based on enhanced MBMS in LTE (Long-Term Evolution), which reduces backhaul traffic. A context-aware architecture for social networking multimedia distribution, which enhances the evolved MBMS systems by adding users' situation knowledge on their assessments and allows personalized services delivery over optimized networks, is proposed in [4]. In [5], the authors describe the design and implementation of an open-source virtualized platform that supports both LTE broadcast services and video streaming services and analyze service performance.

The performance of enhanced MBMS in LTE has been thoroughly examined in previous research works. A comparison of multicast/broadcast services support in LTE-advanced and WiMAX IEEE 802.16m is provided in [6]. Challenges in supporting multicast services over LTE with particular attention to resource management, considered as key aspects for efficient provisioning of MBMS services over cellular networks, are analyzed in [7]. In [8], the authors discuss the relationship between LTE evolved MBMS and next generation broadcast television and propose some recommendations aimed at improving efficiency of the respective systems. In [9], the authors present a compact, convenient model for broadcasting in LTE, as well as a set of efficient algorithms to define broadcasting areas and to actually perform content scheduling. A joint multicast/unicast scheduling strategy for MBMS delivery, which improves QoS performance, is proposed in [10]. In [11], the authors propose a method for forming broadcasting area and assignment of content to them so that radio resources are efficiently exploited and user requests satisfied. In [12], a distributed resource allocation scheme is proposed, which includes device-to-device communication broadcasting groups and minimizes interference relationships. In [13], MBMS with inherently low requirement for network resources as a candidate solution for using such resources in a more efficient manner is proposed. High level considerations related to implementation of a session controller, supporting application programming interfaces for message broadcasting/multicasting, can be found in [14].

The proposed solutions have to be implemented by the telecom network operator. We study another multimedia broadcasting issue that has been scarcely addressed so far, namely, third-party control which allows external applications outside the network to control service provisioning. Our research is focused on two standardized technologies whose synergy has not been studied to date. The Parlay X WS is technology aimed at providing a horizontal service architecture which may be shared by different applications. It defines highly abstracted technology neutral application programming interfaces (APIs). Integration of MBMS and Parlay X requires translation of high level interfaces into telecommunication control protocols. The innovative approach to efficient MBMS provisioning requires considering also other advanced network functions such as policy-based quality of service control and flexible charging. We examine how three different functions in EPS, related to multimedia broadcasting, policy and charging control, and third-party control can collaborate and take the advantage of network convergence in provisioning of value added services.

3. Broadcast Service Architecture in EPS

Different broadcast business cases may be considered and the relevance of each used case is up to network operators and their relationship with the content owners, agencies, and broadcasters. Used case examples include mobile TV, digital radio, video on demand, electronic magazines and newspapers, and configuration and management of intelligent objects with internet connectivity.

As to 3GPP TS 23.246, the components of the 3GPP solution of MBMS are MBMS bearer service and MBMS user service. The MBMS bearer service enables efficient distribution of content to multiple users while MBMS user service is used by user devices when specific end user applications are activated. The architecture of MBMS for EPS with the proposed third-party control is depicted in Figure 1.

The multimedia content that has to be broadcasted is provided by an Application Server (AS). The MBMS AS may be deployed by network operator or third-party application provider. The Broadcast Multicast Service Centre (BMSC) stores multimedia content to be transmitted, controls the multimedia broadcast service, and interacts with media sources and the end users' devices via Packet Data Network Gateway (PDN GW). It provides interfaces for both signaling and data transfer to the MBMS GW. The MBMS Gateway (GW) distributes the data received from the BMSC to the relevant base stations in the access network. The Mobility Management Entity (MME) communicates with the access network in the concerned area, relaying session control information received by MBMS GW [15].

The end users discover the available MBMS services by service announcement, which provides information about services. Service announcement is used to distribute to users information about MBMS, service activation parameters like IP multicast address, and service start time. Service announcement mechanisms may include short message broadcasting, MBMS broadcasting or multicasting mode,

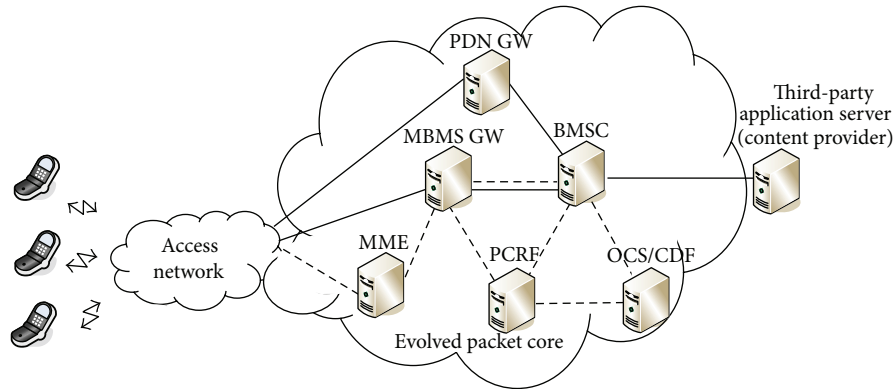


FIGURE 1: Reference architecture of MBMS for EPS.

and a push mechanism. Service announcement details may be found in [16].

An MBMS service may contain multiple distinct multimedia streams and it is provided by session setup and termination. Each session is bundled with bearer establishment and release. Signaling procedures related to MBMS session initiation, update, and termination are specified in [15].

In EPS, the network controls QoS parameters for session broadcast MBMS bearer services. QoS Class Identifier, Allocation and Retention Priority, Maximum Bit Rate (MBR), and Guaranteed Bit Rate (GBR) are applicable to MBMS bearer service, where the MBR is set to be equal to the GBR. The QoS management in EPS is provided by means of policy and charging control (PCC) which allows network operator to authorize and control the usage of resources intended for multimedia traffic [17]. PCC ensures that multimedia services are provided with appropriate transport and charging. The Policy and Charging Rule Function (PCRF) encompasses policy control decision and flow based charging control functions. It receives MBMS session information from BMSC as well as information from the access network and makes policy-based decisions about bearer service session, which are then provided to the MBMS GW. The PCC decisions determine the QoS treatment of the bearer traffic.

Both MBMS content provider and MBMS user may be charged for multimedia broadcasting sessions. EPS provides offline and online charging mechanisms [18]. The BMSC contains triggers that generate events for online and offline charging. The Online Charging System (OCS) performs real-time credit control and its functionality includes transaction handlings, online correlation, and management of user account balances. The Charging Data Function (CDF) receives offline charging information from BMSC and creates call detail records.

4. Parlay X Message Broadcast Web Service

The Parlay X Message Broadcast WS (MBWS) allows third-party applications to send messages to end user devices in a specific geographical area [19]. It provides APIs for sending a broadcast message to the network, for monitoring the delivery status of a sent broadcast message, and for notifications about the message delivery status.

The *sendBroadcastMessage* operation is used to send a broadcast message into the designated area(s).

The *getBroadcastStatus* operation is used by the 3rd-party application to retrieve the status of sent broadcast message. The broadcast status values are as follows:

- (i) *MessageWaiting*. The message is still queued and not delivered to the network yet.
- (ii) *Broadcasting*. The message is being broadcasted as many as requested in the send operation.
- (iii) *Broadcasted*. The message is successfully delivered to network as many as requested.
- (iv) *BroadcastImpossible*. This indicates a final state that delivery of broadcast message is impossible due to specific reasons.
- (v) *BroadcastUnknown*. The message delivery state is unknown.

Figure 2 illustrates the MBMS time line and the message status as seen by 3rd-party application. The *cancelBroadcastMessage* operation may be used by the third-party application to cancel message broadcasting. The operation affects the subsequent broadcast message delivery.

The *notifyBroadcastDeliveryReceipt* operation is used to notify the third-party application about the delivery status of the message. In order to receive notification the third-party application needs to start notifications using *startDeliveryReceiptNotification* operation. The application may end the receipt of notifications using the *stopDeliveryReceiptNotification* operation.

We propose a modified model compared to that presented in [19], which is shown in Figure 3. In *Null* state, there is no message to be broadcasted. As far as the *BroadcastUnknown* state is of temporary kind it is absorbed by the *BroadcastImpossible* and *Broadcasting* states.

5. Model Aspects of Broadcast Multicast Service Centre

5.1. *Modeling Views on Message Broadcast*. BMSC needs to communicate with the third-party application using the

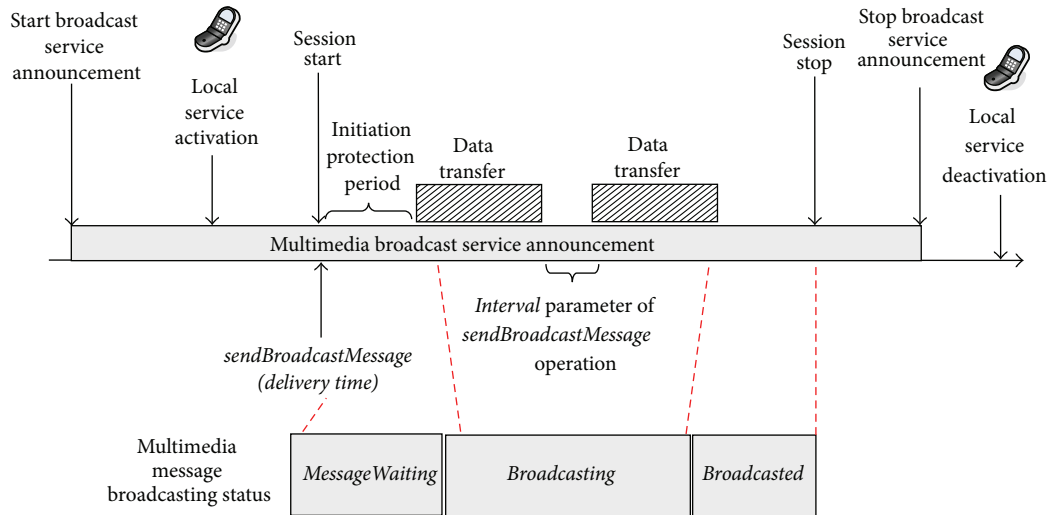


FIGURE 2: Multimedia broadcast service time line and the parameters of *sendBroadcastMessage* operation.

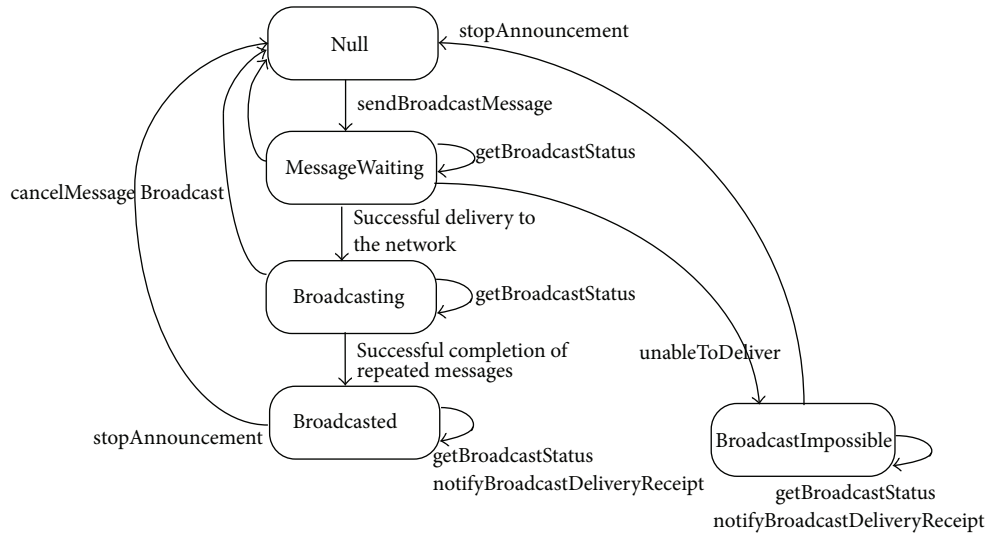


FIGURE 3: Message broadcast state model.

Parlay X MBWS. Toward the network, BMSC has to communicate with MBMS GW to maintain the MBMS session; PCRF to provide information about the MBMS session and to receive notifications about bearers authorized for the session; and CDF to send charging data. Therefore, the BMSC behavior has to be synchronized with the behavior of MBMS GW, PCRF, and CDF, and its view on the MBMS session has to be synchronized with the application view on the state of message broadcast. MBMS GW is in charge of establishment of IP-CAN resources for the MBMS session where IP-CAN stands for Internet Protocol Connectivity Access Network. PCRF controls the QoS that has to be assured for the MBMS session and CDF gathers charging data.

Figure 4 shows the respective state models that have to be synchronized with the BMSC behavior. The BMSC has to transform the MBWS operations and to control

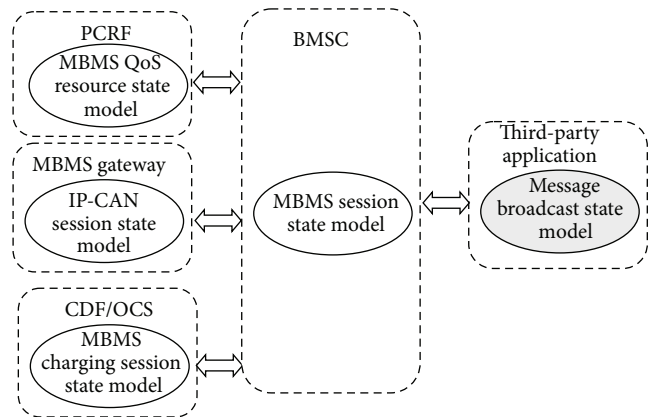


FIGURE 4: Synchronization of MBMS state models in the network and the 3rd-party application view on the message broadcast state.

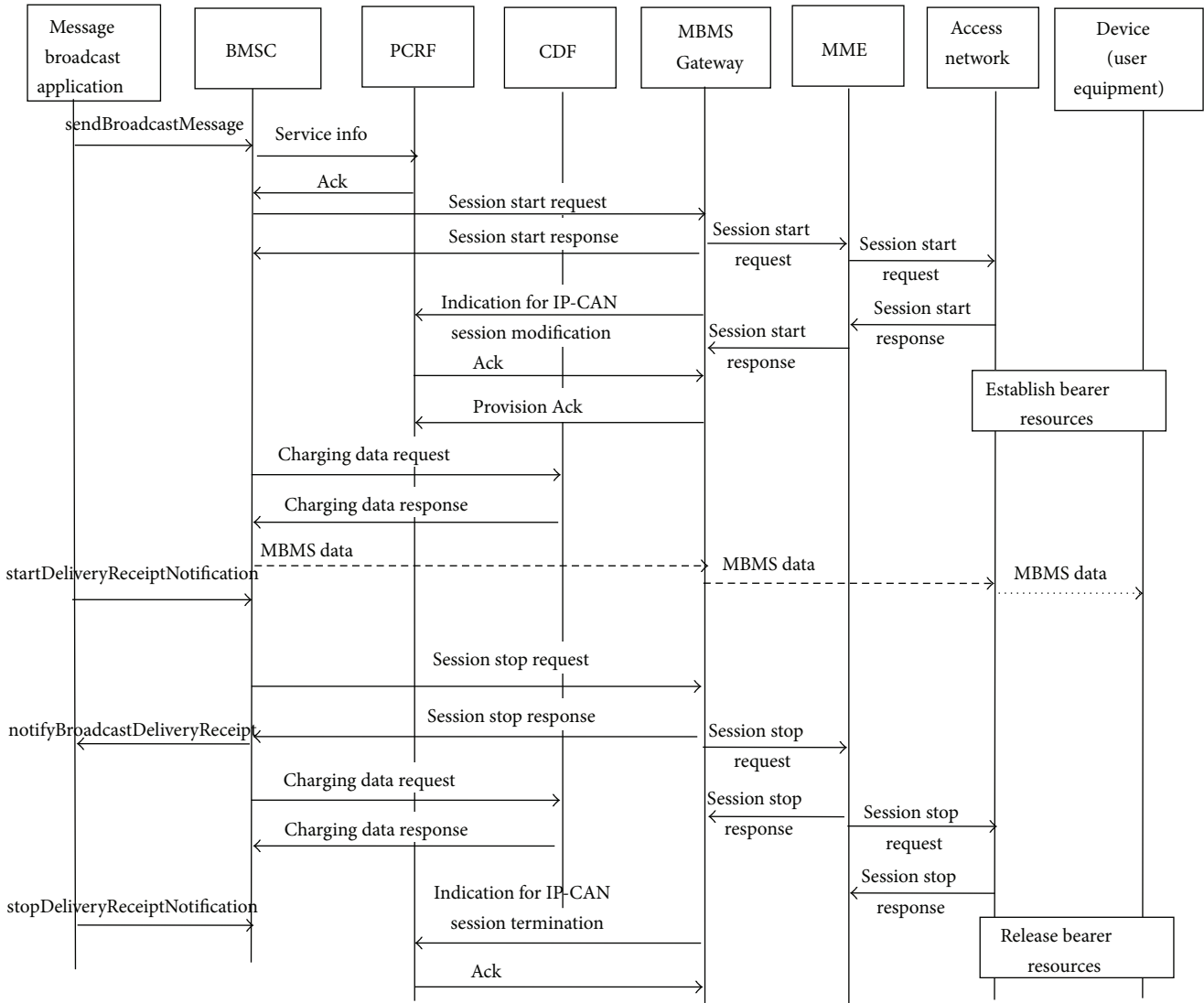


FIGURE 5: Typical message flow illustrating 3rd-party control of MBMS in the network.

the MBMS session appropriately. In practice, the state models of BMSC, MBMS GW, PCRF, and CDF should expose direct correspondence between the transitions and the information they provide; that is, they have to expose equivalent behavior.

The MBMS session state model describes the states of the MBMS session and it is maintained by MBMS GW. The MBMS QoS provisioning requires an MBMS QoS state model reflecting the state of bearer resources allocated by PCRF for the broadcast session. The control protocol used for PCC is Diameter. Similar considerations are applied to the MBMS charging model which holds the states of the MBMS charging session. The control protocol used for charging is also Diameter.

Typical information flows related to multimedia message broadcasting with MBMS session establishment, QoS resource authorization, notification about message broadcast status, and reporting offline charging data are shown in Figure 5.

The MB WS can be used for remote management of intelligent devices like smart meters, which are uniquely identified, gather information from their environment, and communicate that information with network applications. Remote entity management provides means for managing device life cycle including software and firmware upgrade and configuration management. With the increasing number of connected devices, the device management becomes cumbersome and costly for the device operator. So, an EPS operator can sell the MBMS to the device operator to keep the fleet updated. The MBMS can deliver all the software and firmware updates as well as configurations needed to manage the devices up to date.

It is a general assumption that IP connectivity is provided to all devices, which means that for each device there is an established default bearer. Let us assume that the device operator makes use of a message broadcast application. By using the application, the device operator sets the target area,

where the devices are deployed, and writes a message with configuration data. Then the message broadcast application invokes the MB WS which in turn sends a message delivery operation to the BMSC. Subsequently EPS functional entities supporting MBMS deliver the configuration data to all devices in the target area.

When the message broadcast application invokes *sendBroadcastMessage* operation the BMSC opens a Diameter session with the PCRF for the multimedia broadcast session and provides the required session information. The BMSC can subscribe to QoS events related to MBMS bearer resources. The BMSC performs the MBMS session start procedure. The MBMS GW in turn indicates to the PCRF that an IP-CAN session modification is required and waits for PCC rules. Based on session information, the PCRF performs session authorization and sends an acknowledgement with the PCC decisions to the MBMS Gateway. The MBMS starts resources reservation in the access network. When the IP-CAN session modification is completed the MBMS GW sends acknowledgement to the PCRF. The BMSC waits for resource authorization and reservation during the initial protection period and the message broadcast status is *MessageWaiting*. Before starting the MBMS data transfer, the BMSC contacts with the CDF in order to provide charging information. The MBMS content is sent by the BMSC to the MBMS gateway, and then to the respective part of access network. During the MBMS data transfer the message broadcast status is *Broadcasting*. The broadcasted content is received and decoded by the devices (user equipment) that are authorized to join the MBMS service. Using the message broadcast application and the interfaces of the MBWS, the device operator can subscribe for notifications about message delivery (*startDeliveryReceiptNotification* operation) and be notified of a broadcast delivery status of a specific area. Once the MBMS data (configuration data) are broadcasted, the BMSC starts the procedure for session termination, which is followed by release of bearer resources authorized for the MBMS session. The BMSC contacts with the CDF to write the charging data. The message broadcast status becomes *Broadcasted*. The *stopDeliveryReceiptNotification* operation is used to end delivery receipt notification. The content provider is charged for the broadcasted content. The PCRF enables also QoS-based charging. The PCC mechanism supports traffic plane event reporting (e.g., bearer lost) which may reflect in charging. In case of active subscription, the PCRF may notify the BMSC about any QoS events occurring during message broadcasting.

In the course of service provisioning, the BMSC needs to maintain in a synchronized manner three state machines describing the MBMS session, QoS resources authorized for the session, and charging session. These state machines have to be synchronized with the application view on the states of the message broadcast.

On receiving an indication about IP-CAN session modification, the PCRF may reject the requested modification (due to overloading in the respective access network or spending credit limits of the content provider in case of online charging). In this case, the PCRF rejects the IP-CAN session modification requested by the MBMS GW and notifies the

BMSC about the event during the initial protection period. The message broadcast status becomes *BroadcastImpossible*. During message waiting or broadcasting, the device operator may decide to cancel message broadcast. In this case, the third-party application invokes the *cancelBroadcastMessage* operation to request cancelation of the previous *sendBroadcastMessage*. The BMSC in turn initiates session termination toward the MBMS GW and PCRF, which results in IP-CAN session termination and release of QoS resources.

In the next subsection, we provide formal description of the models representing the view on message broadcast of third-party application, BMSC, MBMS GW, PCRF, and CDF, respectively. This will be used to prove formally that the models are synchronized.

5.2. Formal Description of Models Representing Message Broadcast Status. Let us present the state machines as labeled transition systems (LTS).

Definition 1. A labeled transition system (LTS) is a quadruple $(S, Act, \rightarrow, s_0)$, where S is countable set of states, Act is a countable set of elementary actions, $\rightarrow \subseteq S \times Act \times S$ is a set of transitions, and $s_0 \in S$ is the set of initial states.

5.2.1. Application View on Message Broadcast Status. The message broadcast status from application point of view may be *Null*, *MessageWaiting*, *Broadcasting*, or *Broadcasted*. The status is *Null* when there is no message to be broadcasted. The status is *MessageWaiting* if the message is waiting to be delivered to the network. The status is *Broadcasting* or *Broadcasted* if the MBMS data transfer is ongoing or terminated, respectively.

By $T_{App} = (S_{App}, Act_{App}, \rightarrow_{App}, s_0^{App})$ an LTS representing the 3rd-party view on the message broadcast status is denoted where

- (i) $S_{App} = \{\text{Null}, \text{MessageWaiting}, \text{Broadcasting}, \text{Broadcasted}, \text{BroadcastImpossible}\};$
- (ii) $Act_{App} = \{\text{sendMessage}, \text{getBroadcastStatus}, \text{notifyBroadcastDeliveryReceipt}, \text{cancelMessage}, \text{successfulDelivery}, \text{unsuccessfulDelivery}, \text{stopAnnouncement}, \text{unableToDeliver}, \text{successfulMessageCompletion}\};$
- (iii) $\rightarrow_{App} = \{\text{Null} \text{ sendMessage MessageWaiting, MessageWaiting getBroadcastStatus MessageWaiting, MessageWaiting unsuccessfulDelivery Null, MessageWaiting successfulDelivery Broadcasting, MessageWaiting cancelMessage Null, MessageWaiting unableToDeliver BroadcastImpossible, Broadcasting successfulMessageCompletion Broadcasted, Broadcasting cancelMessage Null, Broadcasting getBroadcastStatus Broadcasting, Broadcasted stopAnnouncement Null, Broadcasted getBroadcastStatus Broadcasted, Broadcasted notifyBroadcastDeliveryReceipt Broadcasted, BroadcastImpossible getBroadcastStatus BroadcastImpossible, BroadcastImpossible notifyBroadcastDeliveryReceipt BroadcastImpossible, BroadcastImpossible stopAnnouncement Null}\};$
- (iv) $s_0^{App} = \{\text{Null}\}.$

The *getBroadcastStatus* and *notifyBroadcastDeliveryReceipt* operations do not change the session state or message broadcast status.

5.2.2. BMSC Model of MBMS Session. The simplified BMSC view on the MBMS session states includes the following states. In *MBMSIdle* state, there is no MBMS session. When the application invokes the *sendBroadcastMessage* operation, the BMSC sends session information to the PCRF, a session start request to the MBMS GW, and it moves to *InitialWaiting* state. While being in *InitialWaiting* state, the BMSC waits for resource authorization and QoS resource reservation. The time delay between a session start indication and actual data should be long enough for the required network actions. When the network is ready to send MBMS data, the BMSC sends charging data and moves to *DataTransfer* state. In *DataTransfer* state, MBMS data are transferred to any user equipment which is present. When the MBMS user service determines that there are no more data to send, the BMSC initiates session termination, sends charging data, and moves to *MBMSIdle* state. In *InitialWaiting* or *DataTransfer* state, the BMSC may be notified about any problems related to MBMS bearers and the MBMS session is terminated. It is also possible for third-party application to invoke the *cancelBroadcastMessage* operation while the BMSC is in *InitialWaiting* or *DataTransfer* state, which cancels the MBMS session.

By $T_{\text{BMSC}} = (S_{\text{BMSC}}, Act_{\text{BMSC}}, \rightarrow_{\text{BMSC}}, s_0^{\text{BMSC}})$ an LTS representing the BMSC view on the message broadcast status is denoted where

- (i) $S_{\text{BMSC}} = \{\text{MBMSIdle}, \text{InitialWaiting}, \text{DataTransfer}\};$
- (ii) $Act_{\text{BMSC}} = \{\text{sendBroadcastMessage}, \text{initialPeriod}, \text{dataTransferStop}, \text{resourceFailed}, \text{resourceLost}, \text{cancelBroadcastMessage}\};$
- (iii) $\rightarrow_{\text{BMSC}} = \{\text{MBMSIdle} \xrightarrow{\text{sendBroadcastMessage}} \text{InitialWaiting}, \text{InitialWaiting} \xrightarrow{\text{initialPeriod}} \text{DataTransfer}, \text{InitialWaiting} \xrightarrow{\text{resourceFailed}} \text{MBMSIdle}, \text{InitialWaiting} \xrightarrow{\text{cancelBroadcastMessage}} \text{MBMSIdle}, \text{DataTransfer} \xrightarrow{\text{dataTransferStop}} \text{MBMSIdle}, \text{DataTransfer} \xrightarrow{\text{resourceLost}} \text{MBMSIdle}\};$
- (iv) $s_0^{\text{BMSC}} = \{\text{MBMSIdle}\}.$

Figure 6 illustrates the MBMS session state machine.

5.2.3. MBMS GW Model of IP-CAN Session. The MBMS GW view on the IP-CAN session states for the MBMS user service includes the following states. In *IPCANIdle* state, there is no dedicated IP-CAN session. On receiving session start instructions, the MBMS GW indicates an IP-CAN session modification to the PCRF and moves to *WaitForPCCRules* state. In *WaitForPCCRules* state, the MBMS waits for authorization of IP-CAN session modification. Upon receiving the PCC rules, the MBMS GW initiates IP-CAN session establishment in the access network and moves to *ResourceReservation* state. The MBMS GW moves to *SessionActive* state after it receives an acknowledgement that the resources in the access

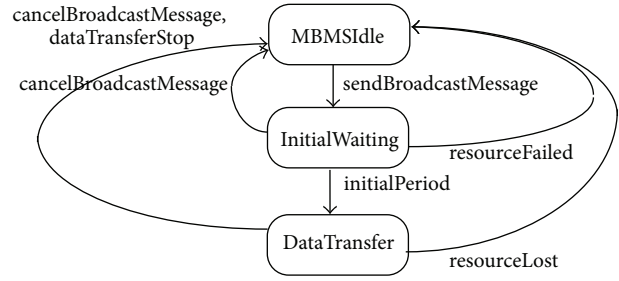


FIGURE 6: MBMS session state machine.

network are reserved. The transition to *ResourceRelease* state occurs when a session stop request is received by the BMSC. In *ResourceRelease* state, the MBMS waits for the release of IP-CAN bearers. In *WaitForTermAck* state, the MBMS GW waits for acknowledgement for session termination from PCRF and then moves to the initial *IPCANIdle* state. In *ResourceReservation* or *SessionActive* state, network, the MBMS GW may receive indications from the PCRF that the bearer modification fails or that the bearers are lost. These cause the transition to *IPCANIdle* state.

By $T_{\text{MBMS}} = (S_{\text{MBMS}}, Act_{\text{MBMS}}, \rightarrow_{\text{MBMS}}, s_0^{\text{MBMS}})$ an LTS representing the MBMS GW view on the MBMS IP-CAN session state is denoted where

- (i) $S_{\text{MBMS}} = \{\text{IPCANIdle}, \text{WaitForPCCRules}, \text{ResourceReservation}, \text{SessionActive}, \text{ResourceRelease}, \text{WaitForTermAck}\};$
- (ii) $Act_{\text{MBMS}} = \{\text{sessionStart}, \text{pccRules}, \text{ipCanBearerEstablishment}, \text{sessionStop}, \text{sessionCancel}, \text{ipCanBearerRelease}, \text{ipCanSessionTermAck}, \text{ipCanBearerFailed}, \text{ipCanBearerLost}\};$
- (iii) $\rightarrow_{\text{MBMS}} = \{\text{IPCANIdle} \xrightarrow{\text{sessionStart}} \text{WaitForPCCRules}, \text{WaitForPCCRules} \xrightarrow{\text{pccRules}} \text{ResourceReservation}, \text{WaitForPCCRules} \xrightarrow{\text{sessionCancel}} \text{IPCANIdle}, \text{ResourceReservation} \xrightarrow{\text{ipCanBearerEstablishment}} \text{SessionActive}, \text{ResourceReservation} \xrightarrow{\text{sessionCancel}} \text{IPCANIdle}, \text{ResourceReservation} \xrightarrow{\text{ipCanBearerFailed}} \text{IPCANIdle}, \text{SessionActive} \xrightarrow{\text{sessionCancel}} \text{ResourceRelease}, \text{SessionActive} \xrightarrow{\text{sessionStop}} \text{ResourceRelease}, \text{SessionActive} \xrightarrow{\text{ipCanBearerLost}} \text{IPCANIdle}, \text{ResourceRelease} \xrightarrow{\text{ipCanBearerRelease}} \text{WaitForTermAck}, \text{WaitForTermAck} \xrightarrow{\text{ipCanSessionTermAck}} \text{IPCANIdle}\};$
- (iv) $s_0^{\text{MBMS}} = \{\text{IPCANIdle}\}.$

The IP-CAN session state machine is depicted in Figure 7.

5.2.4. PCRF Model of MBMS QoS Resources. The PCRF view on the state of QoS resource intended for MBMS session is as follows. In *ResourceNull* state, no QoS resources are authorized for the MBMS session. When the BMSC sends session information, the PCRF performs session binding which takes into account the IP-CAN parameters and moves to *IPCANSession* state storing this information. In *IPCANSession* state, the PCRF may receive an indication about IP-CAN session

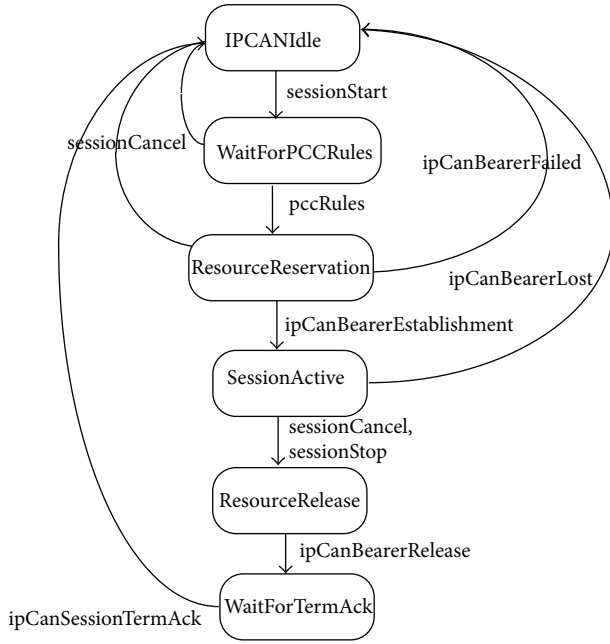


FIGURE 7: IP-CAN session state machine.

modification and correlates appropriate service information with IP-CAN session. In *ResourceAuthorizing* state, the PCRF makes policy decisions and sends them as PCC rules to the MBMS GW. On receiving provision acknowledgement the PCRF moves to *ResourceAuthorized* state. The QoS resources for the MBMS session are authorized and reserved, and the MBMS data transfer can start. When the MBMS GW detects that the IP-CAN session termination is required, it sends an indication to the PCRF. In case of subscription, the PCRF may find the PCC rules that require the BMSC to be notified and sends a notification. The QoS resources authorized for the MBMS session are released which causes the transition to *ResourceNull* state. In *ResourceAuthorizing* state, the PCRF may decide to reject the IP-CAN session modification. Also, during MBMS data transfer any QoS problems may be reported by the MBMS GW to the PCRF, thus affecting the online charging. In case the MBMS GW reports that all bearer resources authorized for the MBMS session are lost, the PCRF moves to *ResourceNull* state.

By $T_{\text{PCRF}} = (S_{\text{PCRF}}, Act_{\text{PCRF}}, \rightarrow_{\text{PCRF}}, s_0^{\text{PCRF}})$ an LTS representing the PCRF GW view on the QoS resource state is denoted where

- (i) $S_{\text{PCRF}} = \{\text{ResourceNull}, \text{IPCanSession}, \text{ResourceAuthorizing}, \text{ResourceAuthorized}\};$
- (ii) $Act_{\text{PCRF}} = \{\text{sessionInfo}, \text{ipCanSessionModification}, \text{provisionAck}, \text{provisionNonAck}, \text{ipCanSessionTermination}, \text{ipCanBearerRemoved}, \text{ipCanSessionCancel}\};$
- (iii) $\rightarrow_{\text{PCRF}} = \{\text{ResourceNull sessionInfo IPCanSession, IPCanSession ipCanSessionModification ResourceAuthorizing, IPCanSession ipCanSessionTermination ResourceNull, ResourceAuthorizing provisionAck ResourceAuthorized, ResourceAuthorizing}$

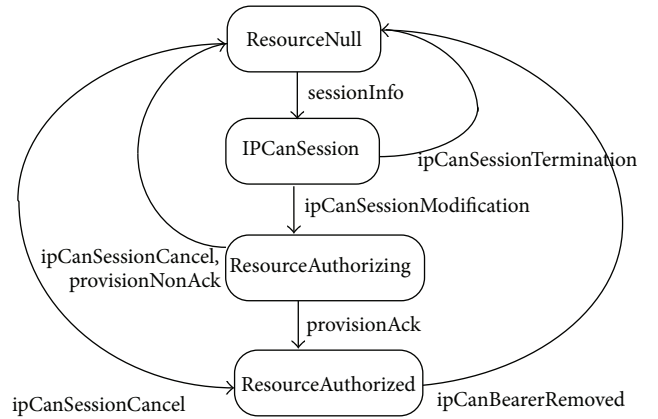


FIGURE 8: QoS resource state machine.

ipCanSessionCancel *ResourceNull*, *ResourceAuthorizing* *provisionNonAck* *ResourceNull*, *ResourceAuthorized* *ipCanSessionCancel* *ResourceNull*, *ResourceAuthorized* *ipCanBearerRemoved* *ResourceNull*};

- (iv) $s_0^{\text{PCRF}} = \{\text{ResourceNull}\}.$

Figure 8 illustrates the QoS resource state machine.

5.2.5. MBMS Charging Session Model. Both offline and online charging are possible. By applying policy and charging control on MBMS sessions, online charging can reflect the QoS experience during message broadcasting. The BMSC collects charging data such as identification of the content source, type of user service, type of bearer resources used to deliver the content broadcast, and identification of users receiving the service. The BMSC collects also charging data such as session duration, data transfer time, and data volume for mobile users receiving the service through MBMS and/or content providers delivery of the MBMS content. The triggers for charging data are bearer service initiation and termination. In *ChargingIdle* state, no charging data are gathered. Charging data are provided by the BMSC upon MBMS session initiation, IP-CAN bearer resource establishment/failure, and MBMS session termination. The charging data are sent to the CDF (in case of offline charging) or to the Online Charging System. On receiving charging data related to session start, the charging session moves to *Active_{initial}* state. In *Active_{initial}* state, when charging data for IP-CAN session establishment are received, the charging session moves to *Active_{transfer}* state. The charging data stop instructions may be received due to normal completion of message broadcast, due to some problems in the network, or due to message broadcast cancellation by the third-party application. On receiving charging data stop instructions the charging session moves to *ChargingIdle* state.

By $T_{\text{CS}} = (S_{\text{CS}}, Act_{\text{CS}}, \rightarrow_{\text{CS}}, s_0^{\text{CS}})$ an LTS representing the charging session states is denoted where

- (i) $S_{\text{CS}} = \{\text{ChargingIdle}, \text{Active}_{\text{initial}}, \text{Active}_{\text{transfer}}\};$

- (ii) $Act_{CS} = \{\text{chargingDataStart}_{\text{init}}, \text{chargingDataStart}_{\text{interim}}, \text{chargingDataStop}_{\text{normal}}, \text{chargingDataStop}_{\text{canceled}}, \text{chargingDataStop}_{\text{failed}}\};$
- (iii) $\rightarrow_{CS} = \{\text{ChargingIdle} \xrightarrow{\text{chargingDataStart}_{\text{init}}} \text{Active}_{\text{initial}}, \text{Active}_{\text{initial}} \xrightarrow{\text{chargingDataStart}_{\text{interim}}} \text{Active}_{\text{transfer}}, \text{Active}_{\text{transfer}} \xrightarrow{\text{chargingDataStop}_{\text{normal}}} \text{ChargingIdle}, \text{Active}_{\text{transfer}} \xrightarrow{\text{chargingDataStop}_{\text{canceled}}} \text{ChargingIdle}, \text{Active}_{\text{transfer}} \xrightarrow{\text{chargingDataStop}_{\text{failed}}} \text{ChargingIdle}, \text{Active}_{\text{initial}} \xrightarrow{\text{chargingDataStop}_{\text{canceled}}} \text{ChargingIdle}, \text{Active}_{\text{initial}} \xrightarrow{\text{chargingDataStop}_{\text{failed}}} \text{ChargingIdle}, \text{Active}_{\text{transfer}} \xrightarrow{\text{chargingDataStop}_{\text{canceled}}} \text{ChargingIdle}, \text{Active}_{\text{transfer}} \xrightarrow{\text{chargingDataStop}_{\text{failed}}} \text{ChargingIdle}\};$
- (iv) $s_0^{CS} = \{\text{ChargingIdle}\}.$

The charging session state machine is depicted in Figure 9.

Having formal description of the models representing message broadcast status as seen by both the third-party application and the network, we can prove that these models are synchronized; that is, they expose equivalent behavior.

6. Formal Verification of Models

6.1. Bisimilarity Concept. Intuitively, in terms of observed behavior, two state machines are equivalent if one state machine displays a final result and the other state machine displays the same result. The idea of equivalence is formalized by the concept of bisimilarity [20]. Strong bisimilarity requires existence of homomorphism between transitions in both state machines. In practice, strong bisimilarity puts strong conditions for equivalence which are not always necessary. For example, internal transitions can present actions, which are intrinsic to the system (i.e., not observable). In weak bisimilarity, internal transitions can be ignored.

The concept of weak bisimilarity is used to study some model aspects of BMSC.

We will use the following notations:

- (i) $s \xrightarrow{a} s'$ stands for the transition (s, a, s') ;
- (ii) $s \xrightarrow{a}$ means that $\exists s' : s \xrightarrow{a} s'$;
- (iii) $s \xRightarrow{\mu} s_n$, where $\mu = a_1, a_2, \dots, a_n : \exists s_1, s_2, \dots, s_n$, such that $s \xrightarrow{a_1} s_1 \xrightarrow{a_2} \dots \xrightarrow{a_n} s_n$;
- (iv) $s \xRightarrow{\mu}$ means that $\exists s'$, such as $s \xRightarrow{\mu} s'$;
- (v) $s \xRightarrow{\bar{\mu}}$ means \Rightarrow if $\mu \equiv \tau$ or $\mu \Rightarrow$, otherwise,

where τ is one or more internal (invisible) actions.

Definition 2. Two labeled transition systems $T = (S, A, \rightarrow, s_0)$ and $T' = (S', A, \rightarrow', s'_0)$ are *weakly bisimilar* ($T \sim T'$) if there is a binary relation $U \subseteq S \times S'$ such that if $s_1 U t_1 : s_1 \subseteq S$ and $t_1 \subseteq S'$, then $\forall a \in Act$:

- (i) $s_1 \xrightarrow{a} s_2$ implies $\exists t_2 : t_1 \xRightarrow{\bar{a}} t_2$ and $s_2 U t_2$;
- (ii) $t_1 \xrightarrow{a'} t_2$ implies $\exists s_2 : s_1 \xRightarrow{\bar{a}} s_2$ and $s_2 U t_2$.

So in order to prove that considered LTS expose equivalent behavior, it is necessary to identify a relation between their states that satisfies the above conditions.

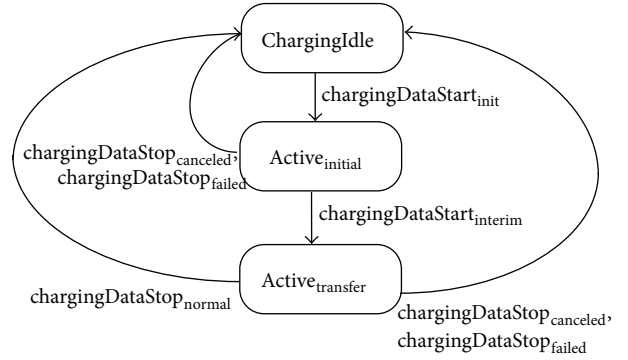


FIGURE 9: Charging session state machine.

6.2. Equivalence of Models Representing Message Broadcast Status

6.2.1. Behavioral Equivalence between Models of IP-CAN Session and QoS Resources. It can be shown that the MBMS GW model of MBMS session and PCRF model of MBMS QoS resources provide the lower level of abstraction, so we will start with considering the synchronization aspects of these models.

Proposition 3. *The labeled transition systems T_{MBMS} and T_{PCRF} are weakly bisimilar.*

Proof. To prove the bisimilarity between two labeled transition systems, it has to be proved that there exists a bisimilar relation between their states. By U_{MP} a relation between the states of T_{MBMS} and T_{PCRF} is denoted where

$$U_{MP} = \{(\text{IpCANIdle}, \text{ResourceNull}), (\text{WaitForPCCRules}, \text{IpCanSession}), (\text{ResourceReservation}, \text{ResourceAuthorizing}), (\text{SessionActive}, \text{ResourceAuthorized})\}.$$

The following homomorphism based on functional mapping may be defined between transitions of T_{MBMS} and T_{PCRF} :

$$\begin{aligned} h_{MP}(\text{sessionStart}) &= h_{MP}(\text{sessionInfo}); \\ h_{MP}(\text{pccRule}) &= h_{MP}(\text{ipCanSessionModification}); \\ h_{MP}(\text{ipCanBearerEstablishment}) &= h_{MP}(\text{provisionAck}); \\ h_{MP}(\text{sessionStop}) &= h_{MP}(\text{ipCanSessionTermination}); \\ h_{MP}(\text{sessionCancel}) &= h_{MP}(\text{ipCanSessionCancel}); \\ h_{MP}(\text{ipCanBearerFailed}) &= h_{MP}(\text{provisionNonAck}); \\ h_{MP}(\text{ipCanBearerLost}) &= h_{MP}(\text{ipCanBearerRemoved}). \end{aligned}$$

The homomorphism between the T_{MBMS} and T_{PCRF} messages shows the action's similarity.

Table 1 represents the functional mapping between the transitions in the IP-CAN session state machine and transitions in the QoS resources state machine.

Based on the established relation between the states of T_{MBMS} and T_{PCRF} , and on the homomorphism h_{MP} between

TABLE 1: Functional mapping between transitions of T_{MBMS} and T_{PCRF} .

Transitions in T_{MBMS}	Transitions in T_{PCRF}	Event
IPCANIdle sessionStart WaitForPCCRules	ResourceNull sessionInfo IPCanSession	MBMS session initiation
WaitForPCCRules pccRules ResourceReservation	IPCanSession ipCanSessionModification ResourceAuthorizing	IP-CAN session modification request
ResourceReservation ipCanBearerEstablishment SessionActive	ResourceAuthorizing provisionAck ResourceAuthorized	IP-CAN session modification authorization
SessionActive sessionStop ResourceRelease, ResourceRelease ipCanBearerRelease WaitForTermAck, WaitForTermAck ipCanSessionTermAck IPCANIdle	ResourceAuthorized ipCanSessionTermination ResourceNull	MBMS session stop
ResourceReservation ipCanBearerFailed IPCANIdle	ResourceAuthorizing provisionNonAck ResourceNull	IP-CAN session modification authorization fails
SessionActive ipCanBearerLost IPCANIdle	ResourceAuthorized ipCanBearerRemoved ResourceNull	All bearers are lost during data transfer
ResourceReservation sessionCancel IPCANIdle	ResourceAuthorized ipCanSessionCancel ResourceNull	The application cancels the MBMS user service during resource reservation
WaitForPCCRules sessionCancel IPCANIdle	IPCANSession ipCanSessionCancel ResourceNull	The application cancels the MBMS user service during resource authorization
SessionActive sessionCancel ResourceRelease, ResourceRelease ipCanBearerRelease WaitForTermAck, WaitForTermAck ipCanSessionTermAck IPCANIdle	ResourceAuthorized ipCanSessionCancel ResourceNull	The application cancels the MBMS user service during data transfer

their transitions, it is proved that $T_{\text{MBMS}} \sim T_{\text{PCRF}}$; that is, they are weakly bisimilar and expose equivalent behavior. \square

6.2.2. Equivalence of Models of MBMS Session, IP-CAN Session, and QoS Resources. A higher level of abstraction on the network resource state for MBMS is provided by the state of MBMS session.

Proposition 4. *The labeled transition systems T_{BMSC} , T_{MBMS} and T_{PCRF} are weakly bisimilar.*

Proof. The weak bisimilarity has the property transitivity; that is, as a relation it is transitive. Then, having proved that $T_{\text{MBMS}} \sim T_{\text{PCRF}}$, it has to be proved that $T_{\text{BMSC}} \sim T_{\text{MBMS}}$.

By U_{BMP} a relation between the states of T_{BMSC} , T_{MBMS} , and T_{PCRF} is denoted where

$$U_{\text{BMP}} = \{(\text{MBMSIdle}, \text{IpCANIdle}), (\text{InitialWaiting}, \text{WaitForPCCRules}), (\text{DataTransfer}, \text{SessionActive})\}.$$

The following homomorphism based on functional mapping may be defined between transitions of T_{BMSC} , T_{MBMS} , and T_{PCRF} :

$$h_{\text{BMP}}(\text{sendBroadcastMessage}) = h_{\text{BMP}}(\text{sessionz-Start}) = h_{\text{BMP}}(\text{sessionInfo});$$

$$h_{\text{BMP}}(\text{initialPeriod}) = h_{\text{BMP}}(\text{pccRule}) = h_{\text{BMP}}(\text{ip-CanSessionModification});$$

$$h_{\text{BMP}}(\text{dataTransferStop}) = h_{\text{BMP}}(\text{sessionStop}) = h_{\text{BMP}}(\text{ipCanSessionTermination});$$

$$h_{\text{BMP}}(\text{cancel}) = h_{\text{BMP}}(\text{sessionCancel}) = h_{\text{BMP}}(\text{ip-CanSessionCancel});$$

$$h_{\text{BMP}}(\text{resourcesFailed}) = h_{\text{BMP}}(\text{ipCanBearerFailed}) = h_{\text{BMP}}(\text{provisionNonAck});$$

$$h_{\text{BMP}}(\text{resourcesLost}) = h_{\text{BMP}}(\text{ipCanBearerLost}) = h_{\text{BMP}}(\text{ipCanBearerRemoved}).$$

TABLE 2: Functional mapping between transitions of T_{BMSC} and T_{MBMS} .

Transitions in T_{BMSC}	Transitions in T_{MBMS}	Event
MBMSIdle sendBroadcastMessage InitialWaiting	IPCANIdle sessionStart WaitForPCCRules	MBMS session initiation
InitialWaiting initialPeriod DataTransfer	WaitForPCCRules pccRules ResourceReservation, ResourceReservation ipCanBearerEstablishment SessionActive	IP-CAN session modification request IP-CAN session modification authorization
DataTransfer dataTransferStop MBMSIdle	SessionActive sessionStop ResourceRelease, ResourceRelease ipCanBearerRelease WaitForTermAck, WaitForTermAck ipCanSessionTermAck IPCANIdle	MBMS session stop
InitialWaiting resourceFailed MBMSIdle	ResourceReservation ipCanBearerFailed IPCANIdle	IP-CAN session modification authorization fails
DataTransfer resourceLost MBMSIdle	SessionActive ipCanBearerLost IPCANIdle	All bearers are lost during data transfer
InitialWaiting cancelBroadcastMessage MBMSIdle	WaitForPCCRules sessionCancel IPCANIdle ResourceReservation sessionCancel IPCANIdle	The application cancels the MBMS user service during resource reservation
DataTransfer cancelBroadcastMessage MBMSIdle	SessionActive sessionCancel ResourceRelease, ResourceRelease ipCanBearerRelease WaitForTermAck, WaitForTermAck ipCanSessionTermAck IPCANIdle	The application cancels the MBMS user service during resource authorization

The functional mapping between the transitions in the MBMS session and IP-CAN state machines represented in Table 2 shows that U_{BMP} is a bisimilar relation.

From the homomorphism h_{BMP} and bisimilar property of U_{BMP} it follows that $T_{\text{BMSC}} \sim T_{\text{MBMS}}$.

From the bisimulation transitivity it follows that $T_{\text{BMSC}} \sim T_{\text{MBMS}} \sim T_{\text{PCRF}}$.

This means that MBMS session state machine, IP-CAN state machine, and the QoS resources state machine are weakly bisimilar. \square

6.2.3. Equivalence of Application and Network Models. The highest level of abstraction on message broadcast status is provided by 3rd-party application. Its view has to be synchronized also with the network resource state and charging session state.

Proposition 5. *The labeled transition systems T_{App} , T_{BMSC} , T_{MBMS} , T_{PCRF} , and T_{CS} are weakly bisimilar.*

Proof. By U_{ABMPC} a relation between the states of T_{App} , T_{BMSC} , T_{MBMS} , T_{PCRF} , and T_{CS} is denoted where

$$U_{\text{ABMPC}} = \{(\text{Null}, \text{MBMSIdle}, \text{IpCANIdle}, \text{ResourceNull}, \text{ChargingIdle}), (\text{MessageWaiting}, \text{InitialWaiting}, \text{WaitForPCCRules}, \text{IpCanSession}, \text{Active}_{\text{initial}}), (\text{Broadcasting}, \text{DataTransfer}, \text{SessionActive}, \text{ResourceAuthorized}, \text{Active}_{\text{transfer}})\}.$$

The following homomorphism based on functional mapping may be defined between transitions of T_{App} , T_{BMSC} , T_{MBMS} , T_{PCRF} , and T_{CS} :

$$\begin{aligned} h_{\text{ABMPC}}(\text{sendMessage}) &= h_{\text{ABMPC}}(\text{sendBroadcastMessage}) = h_{\text{ABMPC}}(\text{sessionStart}) = h_{\text{ABMPC}}(\text{sessionInfo}) = h_{\text{ABMPC}}(\text{sendBroadcastMessage}) = h_{\text{ABMPC}}(\text{chargingDataStart}_{\text{init}}); \end{aligned}$$

$$\begin{aligned} h_{\text{ABMPC}}(\text{successfulDelivery}) &= h_{\text{ABMPC}}(\text{initialPeriod}) = h_{\text{ABMPC}}(\text{pccRule}) = h_{\text{ABMPC}}(\text{ipCanSessionModification}) = h_{\text{ABMPC}}(\text{chargingDataStart}_{\text{interim}}); \end{aligned}$$

$$\begin{aligned}
h_{ABMPC}(\text{successfulMessageCompletion}) &= \\
h_{ABMPC}(\text{dataTransferStop}) &= h_{ABMPC}(\text{sessionStop}) = \\
h_{ABMPC}(\text{ipCanSessionTermination}) &= h_{ABMPC}(\text{chargingDataStop}_{\text{normal}}); \\
h_{ABMPC}(\text{cancelMessage}) &= h_{ABMPC}(\text{cancel}) = \\
h_{ABMPC}(\text{sessionCancel}) &= h_{ABMPC}(\text{ipCanSessionCancel}) = h_{ABMPC}(\text{chargingDataStop}_{\text{canceled}}); \\
h_{ABMPC}(\text{unsuccessfulDelivery}) &= h_{ABMPC}(\text{resourceFailed}) = \\
h_{ABMPC}(\text{provisionNonAck}) &= h_{ABMPC}(\text{chargingDataStop}_{\text{failed}}).
\end{aligned}$$

In order to define a weak bisimilar relation between the states of T_{App} , T_{BMS} , T_{MBMS} , T_{PCRF} , and T_{CS} we establish the valid sequences of ordered transitions from $- \rightarrow_{App}$, $- \rightarrow_{BMS}$, $- \rightarrow_{MBMS}$, $- \rightarrow_{PCRF}$, and $- \rightarrow_{CS}$ that correspond to U_{ABMPC} .

Let $\tau_i^{App} \subset \rightarrow_{App}$, where $i \in \{1, \dots, 6\}$, be the following transition sequences:

$$\begin{aligned}
\tau_1^{App} &= (\text{Null sendMessage MessageWaiting}); \\
\tau_2^{App} &= (\text{MessageWaiting getBroadcastStatus MessageWaiting, MessageWaiting successfulDelivery Broadcasting}); \\
\tau_3^{App} &= (\text{Broadcasting getBroadcastStatus Broadcasting, Broadcasting successfulMessageCompletion Broadcasted, Broadcasted getBroadcastStatus Broadcasted, Broadcasted notifyBroadcastDeliveryReceipt Broadcasted, Broadcasted stopAnnouncement Null}); \\
\tau_4^{App} &= (\text{MessageWaiting unableToDeliver BroadcastImpossible, BroadcastImpossible getBroadcastStatus BroadcastImpossible, BroadcastImpossible notifyBroadcastDeliveryReceipt BroadcastImpossible, BroadcastImpossible stopAnnouncement Null}); \\
\tau_5^{App} &= (\text{MessageWaiting cancelMessage Null}); \\
\tau_6^{App} &= (\text{Broadcasting cancelMessage Null}).
\end{aligned}$$

Let $\tau_i^{BMS} \subset \rightarrow_{BMS}$, where $i \in \{1, \dots, 6\}$, be following transition sequences:

$$\begin{aligned}
\tau_1^{BMS} &= (\text{MBMSIdle sendBroadcastMessage InitialWaiting}); \\
\tau_2^{BMS} &= (\text{InitialWaiting initialPeriod DataTransfer}); \\
\tau_3^{BMS} &= (\text{DataTransfer dataTransferStop MBMSIdle}); \\
\tau_4^{BMS} &= (\text{InitialWaiting resourceFailed MBMSIdle}); \\
\tau_5^{BMS} &= (\text{InitialWaiting cancelBroadcastMessage MBMSIdle}); \\
\tau_6^{BMS} &= (\text{DataTransfer resourceLost MBMSIdle}).
\end{aligned}$$

Let $\tau_i^{MBMS} \subset \rightarrow_{MBMS}$, where $i \in \{1, \dots, 6\}$, be the following:

$$\begin{aligned}
\tau_1^{MBMS} &= (\text{IPCANIdle sessionStart WaitForPCRules}); \\
\tau_2^{MBMS} &= (\text{WaitForPCCRules pccRules ResourceReservation, ResourceReservation ipCanBearerEstablishment SessionActive});
\end{aligned}$$

$$\tau_3^{MBMS} = (\text{SessionActive sessionStop ResourceRelease, ResourceRelease ipCanBearerRelease WaitForTermAck, WaitForTermAck ipCanSessionTerminationAck IPCANIdle});$$

$$\tau_4^{MBMS} = (\text{ResourceReservation ipCanBearerFailed IPCANIdle});$$

$$\tau_5^{MBMS} = (\text{WaitForPCCRules sessionCancel IPCANIdle, ResourceReservation sessionCancel IPCANIdle});$$

$$\tau_6^{MBMS} = (\text{SessionActive ipCanBearerLost IPCANIdle}).$$

Let $\tau_i^{PCRF} \subset \rightarrow_{PCRF}$, where $i \in \{1, \dots, 6\}$, be the following:

$$\begin{aligned}
\tau_1^{PCRF} &= (\text{ResourceNull sessionInfo IPCANSession}); \\
\tau_2^{PCRF} &= (\text{IPCANSession ipCanSessionModification ResourceAuthorizing, ResourceAuthorizing provisionAck ResourceAuthorized}); \\
\tau_3^{PCRF} &= (\text{ResourceAuthorized ipCanSessionTermination ResourceNull}); \\
\tau_4^{PCRF} &= (\text{ResourceAuthorizing provisionNonAck ResourceNull}); \\
\tau_5^{PCRF} &= (\text{IPCANSession ipCanSessionCancel ResourceNull, ResourceAuthorized ipCanSessionCancel ResourceNull}); \\
\tau_6^{PCRF} &= (\text{ResourceAuthorized ipCanBearerRemoved ResourceNull}).
\end{aligned}$$

Let $\tau_i^{CS} \subset \rightarrow_{CS}$, where $i \in \{1, \dots, 6\}$, be following transition sequences:

$$\begin{aligned}
\tau_1^{CS} &= (\text{ChargingIdle chargingDataStart}_{\text{init}} \text{Active}_{\text{initial}}); \\
\tau_2^{CS} &= (\text{Active}_{\text{initial}} \text{chargingDataStart}_{\text{interim}} \text{Active}_{\text{transfer}}); \\
\tau_3^{CS} &= (\text{Active}_{\text{transfer}} \text{chargingDataStop}_{\text{normal}} \text{ChargingIdle}); \\
\tau_4^{CS} &= (\text{Active}_{\text{initial}} \text{chargingDataStop}_{\text{failed}} \text{ChargingIdle}); \\
\tau_5^{CS} &= (\text{Active}_{\text{initial}} \text{chargingDataStop}_{\text{canceled}} \text{ChargingIdle}); \\
\tau_6^{CS} &= (\text{Active}_{\text{transfer}} \text{chargingDataStop}_{\text{canceled}} \text{ChargingIdle}).
\end{aligned}$$

Then, based on the defined homomorphism h_{ABMPC} between the actions of T_{App} , T_{BMS} , T_{MBMS} , T_{PCRF} , and T_{CS} , it follows that for U_{ABMPC} the sequences $(\tau_i^{App}, \tau_i^{BMS}, \tau_i^{MBMS}, \tau_i^{PCRF}, \tau_i^{CS})$ for $i \in \{1, \dots, 6\}$ agree with Definition 2 for weak bisimilarity.

Hence, the state machines that represent the 3rd-party application view on message broadcast, MBMS session, IP-CAN session, QoS resources, and charging session are bisimilar; that is, these state machines expose equivalent behavior. \square

7. Conclusion

In EPS there are specific features, nodes, and interfaces defined to support broadcasting of content to multiple users simultaneously. Parlay X Message Broadcast WS allows third-party applications to send messages to all mobile terminals in a specific geographical area. The deployment of 3rd-party control on multimedia broadcasting implies more research on synergy between WS APIs and respective communication control protocols.

The paper presents a study on modeling aspects of a network node that mediates between 3rd-party application servers and the evolved packet core of a mobile network. It explores the way the network agnostic WS interfaces collaborate with specific control functionality in order to provide value added multimedia broadcasting services.

The results regard model aspects of the behavior of a Service Capability Server, namely, BMSC, which exposes Parlay X WS interfaces toward applications and “talks” network protocols toward the network. The focus is on collaboration of different functionalities, namely, service control, session management, QoS control, and charging control. In the context of message broadcast, we modeled the states of MBMS session, IP-CAN session, QoS resources, and charging session and showed that these models are synchronized with the 3rd-party application view on message broadcast. The proof is based on functional mapping between the transitions in state machines and the identification of a homomorphism between actions. The concept of weak bisimilarity is used to prove that the state machines expose equivalent behavior. Thus, the novelty of contribution might be summarized as model definitions and proof for bisimilarity.

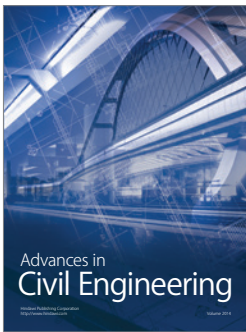
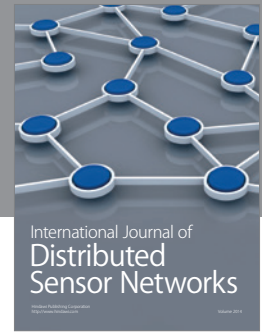
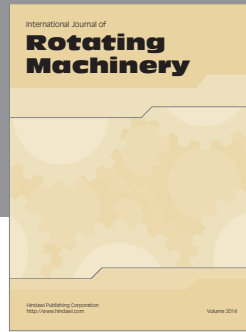
The utilization of standardized Message Broadcast APIs provides a unified approach for 3rd-party application management and provisioning. It allows development of converged multimedia broadcast applications in a way which is independent of the underlying access technologies.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

- [1] D. Lecompte and F. Gabin, “Evolved multimedia broadcast/multicast service (eMBMS) in LTE-advanced: overview and Rel-11 enhancements,” *IEEE Communications Magazine*, vol. 50, no. 11, pp. 68–74, 2012.
- [2] J. Calabuig, J. F. Monserrat, D. Gozávez, and O. Klemp, “Safety on the roads: LTE alternatives for sending ITS messages,” *IEEE Vehicular Technology Magazine*, vol. 9, no. 4, pp. 61–70, 2014.
- [3] I. Stephanasik, I. Chochliouros, G. Lymperopoulos, and K. Berberidis, “Optimal video delivery in mobile networks using a cache-accelerated multi area eMBMS architecture,” in *Artificial Intelligence Applications and Innovations*, vol. 437 of *IFIP Advances in Information and Communication Technology*, pp. 13–23, Springer, Berlin, Germany, 2014.
- [4] F. C. Pinto, N. Carapeto, A. Videira, T. Frazão, and M. Homem, “Context-aware multimedia distribution to mobile social communities,” *International Journal of Handheld Computing Research*, vol. 4, no. 3, pp. 63–92, 2013.
- [5] C. M. Lentisco, M. Aguayo, L. Bellido, E. Pastor, D. De-Antonio-Monte, and A. G. Bolivar, “A virtualized platform for analyzing LTE broadcast services,” in *Proceedings of the European Conference on Networks and Communications (EuCNC '15)*, pp. 512–516, Paris, France, June 2015.
- [6] J. Calabuig, J. Monserrat, D. Martin-Sacristan, and J. Olmos, “Comparison of multicast/broadcast services in Long Term Evolution Advanced and IEEE 802.16 m networks,” *Wireless Communications and Mobile Computing*, vol. 14, no. 7, pp. 717–728, 2014.
- [7] G. Araniti, M. Condoluci, and A. Molinaro, “Resource management of multicast services over LTE,” in *Convergence of Broadband, Broadcast, and Cellular Network Technologies*, pp. 77–93, IGI Global, 2014.
- [8] G. K. Walker, J. Wang, C. Lo, X. G. Zhang, and G. Bao, “Relationship between LTE broadcast/eMBMS and next generation broadcast television,” *IEEE Transactions on Broadcasting*, vol. 60, no. 2, pp. 185–192, 2014.
- [9] F. Malandrino, C. Casetti, C. F. Chiasserini, and S. Zhou, “Real-time scheduling for content broadcasting in LTE,” in *Proceedings of the IEEE International Symposium on Modelling, Analysis & Simulation of Computer and Telecommunication Systems*, pp. 126–131, Paris, France, September 2014.
- [10] A. de la Fuente, A. G. Armada, and R. P. Leal, “Joint multicast/unicast scheduling with dynamic optimization for LTE multicast service,” in *Proceedings of the 20th European Wireless Conference (EW '14)*, pp. 462–467, Barcelona, Spain, May 2014.
- [11] C. Borgiattino, C. Casetti, C. F. Chiasserini, and F. Malandrino, “Efficient area formation for LTE broadcasting,” in *Proceedings of the 12th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON '15)*, pp. 202–210, IEEE, Seattle, Wash, USA, June 2015.
- [12] C. W. Yeh, M. J. Shih, G. Y. Lin, and H. Y. Wei, “LTE-D broadcast with distributed interference-aware D2D resource allocation,” in *Proceedings of the 7th International Conference on Ubiquitous and Future Networks (ICUFN '15)*, pp. 165–170, Sapporo, Japan, July 2015.
- [13] Y.-H. Xu, C.-O. Chow, M.-L. Tham, and H. Ishii, “An enhanced framework for providing multimedia broadcast/multicast service over heterogeneous networks,” *Journal of Zhejiang University Science*, vol. 15, no. 1, pp. 63–80, 2014.
- [14] J. H. Kim, S. K. Kim, and B. S. Lee, “Message service method and message service system,” US Patent 8478313 B2, 2013.
- [15] 3GPP, “Multimedia Broadcast/Multicast Service (MBMS); architecture and functional description,” 3GPP TS 23.246, 2015, Release 13.
- [16] 3GPP, “Multimedia Broadcast/Multicast Service (MBMS); protocols and codecs,” 3GPP TS 26.346, 2015, Release 13.
- [17] 3GPP TS 23.203, “Policy and Charging control architecture,” Release 13, v13.5.0, 2015.
- [18] 3GPP, “Telecommunication management; charging management; charging architecture and principles,” 3GPP TS 32.240, 2015, Release 13, v13.0.0.
- [19] 3GPP, “Open Service Access (OSA); parlay X web services; part 15: message broadcast,” 3GPP TS 29.199-15, 2009, Release 9.
- [20] L. Fuchun, Z. Qiansheng, and C. Xuesong, “Bisimilarity control of decentralized nondeterministic discrete-event systems,” in *Proceedings of the 33rd Chinese Control Conference (CCC '14)*, pp. 3898–3903, IEEE, Nanjing, China, July 2014.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

