REAL-TIME MULTI-USER TRANSCODING FOR PUSH TO TALK OVER CELLULAR

Stéphane Coulombe[1]

École de technologie supérieure, Department of Software and IT Engineering
1100 Notre Dame Ouest, Montreal, Qc, Canada, H3C 1K3
e-mail: stephane.coulombe@etsmtl.ca

ABSTRACT

Transcoding is required to enable interoperability between Push to talk over Cellular (PoC) clients with incompatible capabilities (e.g. between a PoC client supporting the AMR speech codec and another supporting EVRC). Although the Open Mobile Alliance (OMA) recognizes the need for transcoding in the PoC application, no solution is provided by the standard to enable it. In this paper, we present a transcoding system for real-time multi-user PoC sessions. The solution is centralized at the Controlling PoC Function which manages session control operations and the flow of media streams to enable transcoding to be performed in a distinct transcoding server (TS). There are several advantages to this solution, such as scalability, applicability to all PoC group session scenarios, compatibility with existing PoC specifications, and transparency for existing PoC clients.

Index Terms— Transcoding, multi-user, PoC, IMS.

1. INTRODUCTION

The Push to talk over Cellular (PoC) service allows mobile users to create group sessions where participants can engage in voice and data communications on a 1-to-1 or 1-to-many basis [1], as illustrated in Figure 1. The voice communications are similar to those of walkie-talkie services, where terminals have dedicated 'talk' buttons. Only one person can speak at any given time, and each Talk Burst (TB) is relatively short (a few seconds). Each TB is copied to all the other participants in the session. Users can also exchange instant messages (IMs). Soon, voice-only TBs will evolve into voice and video TBs, and IMs will contain rich media content (audio, video frames, text, etc.).

Because of the diversity of terminals and networks, issues associated with interoperability are arising. For instance, 3GPP mandates the AMR narrowband speech codec as the default speech codec for the PoC service [2]. Further, 3GPP mandates support of the AMR wideband speech codec, if the PoC client's equipment uses a 16 kHz sampling frequency for speech. In contrast, 3GPP2 mandates the EVRC speech codec as the default speech codec [3]. More serious incompatibilities are expected to arise with respect to video streams (with various codecs, such as H.263, MPEG-4, and H.264) and media rich IMs.

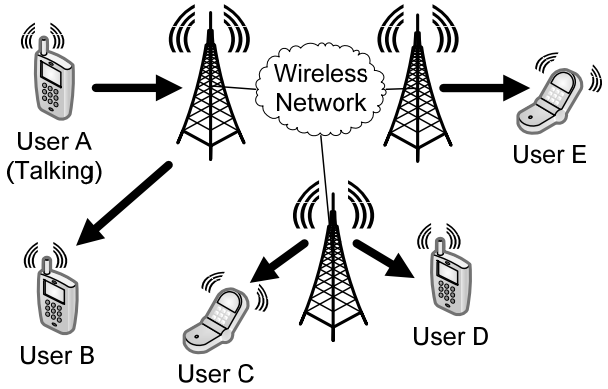

Figure 1: Example of a 1-to-many group session (voice).

In the PoC standard (versions 1.0 and 2.0), the need for transcoding is well recognized, but no detailed solution is provided. It is stated in [1] that transcoding may be performed by both the Controlling PoC Function (CPF) and the Participating PoC Function (PPF), although the means for achieving this is not provided. In PoC version 2.0 [4], the PoC Interworking Function has been introduced, which may, among other things, perform transcoding. But its realization is outside the scope of the OMA, and currently no solution has been proposed by the standards bodies to ensure interoperability in PoC. Furthermore, the author is not aware of any solution proposed to enable interoperability in SIP-based real-time multi-party sessions (as found in PoC), besides the obvious and inefficient Back-to-Back User Agent (B2BUA)-based approaches [5].

This paper proposes a solution to support transcoding within the scope of the real-time multi-user sessions offered by PoC. In the proposed solution, transcoding will be managed by the CPF and performed in a separate logical entity, the Transcoding Server (TS). We will show that actions must be taken at different levels of the session to enable transcoding:

[1] This work was funded by Vantrix Corp. (http://vantrix.com/).

session offering, session control, and media control. An important feature of the proposed solution is that it is compatible with the existing PoC architecture and protocols.

The paper is organized as follows. In section 2, we present an overview of the PoC architecture; in section 3, the proposed solution; in section 5, the advantages and disadvantages of the solution, as well as other possible solutions; and in section 6 our conclusions.

2. OVERVIEW OF THE POC ARCHITECTURE

We assume that the reader is fairly familiar with the PoC system, as described in [1, 6, 7]. The overall PoC architecture, enhanced with the TS, for the generic case of users distributed over different networks is illustrated in Figure 2. Here, we see various PoC clients, each connected to its own PPF (over its own network), participating in a common session controlled by a CPF. The PoC service is built on top of a SIP/IP core, which could correspond to the 3GPP IP Multimedia Sub-system (IMS) [8, 9] or to the 3GPP2 IMS [10, 11].

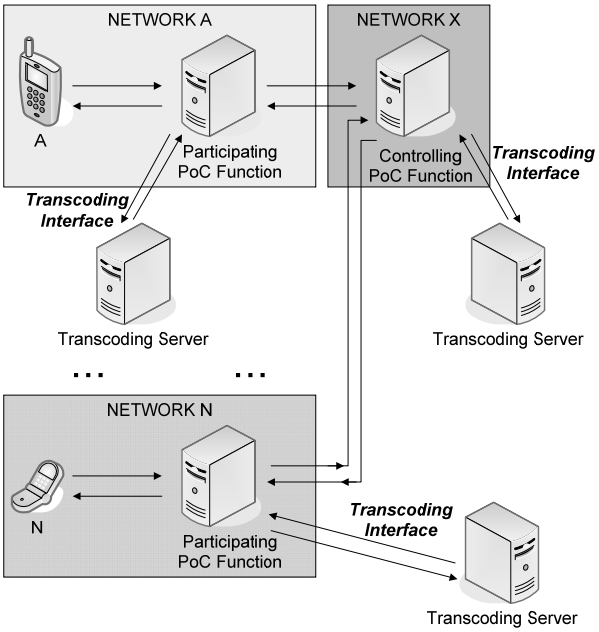

Figure 2: High-level architecture of the PoC application with transcoding.

The CPF provides centralized PoC session handling, which includes RTP media distribution (copies of RTP packets to each participant), Talk Burst Control (TBC), policy enforcement for participation in group sessions, and the participant information. The PPF provides PoC session handling (such as policy enforcement for incoming PoC sessions) and relays TBC messages (to manage who has permission to speak) between the PoC client and the CPF. It may also relay RTP media between the PoC client and the CPF.

It is important to note that the CPF is responsible for managing (deciding) who has permission to speak at any given time and for copying media packets from the source to the other participants. The PPF cannot perform these operations. Note that, at any given time, at most one user has permission to speak. The request to speak is made when the user presses the 'talk' button on his mobile. The TB will last until the user releases the button, at which time a TB Complete message is sent to the CPF.

In principle, transcoding can be centrally managed by the CPF or distributed among the various PPFs. We maintain that managing it centrally at the CPF is more efficient, however, because the CPF has a global view of the session. Indeed, it 'knows' who has permission to speak and is responsible for duplicating the media packets. As a result, it can manage the transcoding operations optimally, in comparison to a PPF, which has only a local view of the session (further justification for this choice will be presented in section 4). We therefore focus on the management of transcoding at the CPF.

3. TRANSCODING CENTRALIZED AT THE CPF

This section describes the various elements required to support transcoding centralized at the CPF.

3.1. Roles of the CPF

The CPF must manage the transcoding operations in addition to managing permission to speak. The CPF has two main responsibilities with respect to enabling transcoding:

1. Manage session control operations (setup and update):
   - Setup: ensure that mobile devices with incompatible capabilities (codecs) will nevertheless transparently connect together in a session, and manage the transcoding operations to be performed by the TS.
   - Update: update transcoding operations as different users have permission to speak or as users join and leave a session.

2. Manage the flow of media streams between users:
   - When transcoding is required, the media streams (RTP packets) will have to flow through a TS, where they will be transcoded and then sent to their destination. This requires that the media flow be managed by the CPF.

The sub-sections below will explain how these roles can be fulfilled.

3.2. Session control managed by the CPF

During the session setup phase, as PoC clients may support incompatible codecs, the CPF may have to change the Session Description Protocol (SDP) [12] by adding to the SDP list the codecs supported by that client, and for which a proper transcoding to the codecs of other participating clients is possible. For instance, a PoC client supporting only AMR would not normally be able to establish a direct session with a client supporting EVRC (see Figure 3a). However, a CPF enabling AMR-EVRC transcoding would include both EVRC and AMR among the session offerings, as illustrated in Figure 3b.

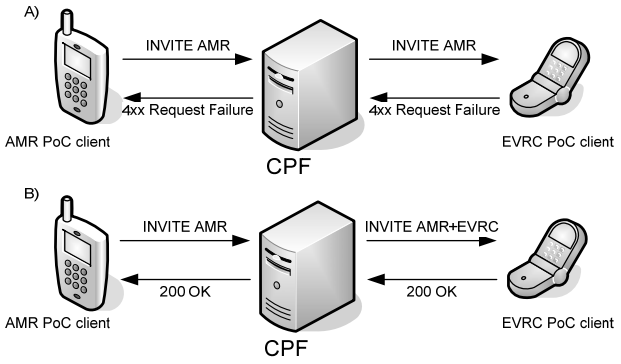

Figure 3: CPF role of ensuring proper session offerings: a) CPF not supporting transcoding; b) CPF supporting transcoding and enhancing codec offerings.

This operation is not as straightforward at it looks, since not only must new codecs be added to the SDP, but new IP addresses and ports must be provided in the invitations, in order for the media flows to be rerouted through the TS.

Figure 4 shows an example of the control flow between the CPF, the TS, and the PoC clients when a session is set up with transcoding enabled. When client A invites another user to speak, the CPF first asks the TS to set up a transcoding session and requests a list of acceptable codecs to offer to other users. The CPF forwards the enhanced invitation to the other client, which accepts with a different codec from that of client A. The CPF then updates the transcoding session by, among other things, providing information about the selected codec. The CPF informs client A that the invitation has been accepted with the codec offered. Although this is not illustrated, it is assumed that the user of client A has obtained permission to speak. This system then proceeds to send AMR packets to the TS, which transcodes them to EVRC and forwards them to client B. For this to happen, the IP addresses and ports in the SDP messages are modified during the invitation process (either by the CPF or by the TS), in order for users to send packets through the TS. The TS is also informed about the addresses and ports of each participant, as well as the capabilities they support (codecs).

This whole process is totally transparent to PoC clients, which is an important feature of the proposed solution, as there are already many PoC clients in use.

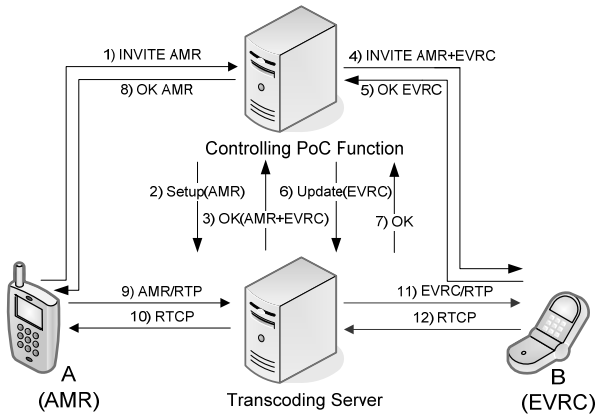

Figure 4: Example of session control flow for transcoding centralized at the CPF. All media packets arrive at the TS.

Even once the session has been set up, the CPF has to continually manage the session, updating transcoding operations to be performed by the TS, as well as the media flow. Indeed, when the session parameters change (e.g. to account for a joining or departing client) or when a different user has permission to speak, the CPF will have to inform the TS of the situation so that proper transcoding and routing of streams will be performed (indeed, the media packets have to be sent to every user except the one who has permission to speak). For instance, let us consider a session where participating clients support either AMR or EVRC. If the PoC device of the user with permission to speak supports AMR, then AMR to EVRC transcoding is performed. However, if that user's device supports EVRC, then EVRC to AMR transcoding is performed. Also, the list of destinations changes, based on who has permission to speak.

In Figure 5, we can see an example of the control flow between the CPF, the TS, and the clients when a user requests permission to speak. We assume that initially no one has permission to speak. The user of client A asks for permission to speak by issuing a TB request (we assume the media flow passing through the TS is as described in option 2 of Figure 7 -- to be discussed in the next subsection). The request arrives at the TS and is forwarded to the CPF. The CPF informs the TS that the user now has permission to speak so that the latter can allocate transcoding resources properly and enforce proper control over streams. When the TS confirms that the request has been granted, the CPF

informs the client that this is the case. Client A can then start sending AMR packets, which are transcoded prior to being sent to client B.

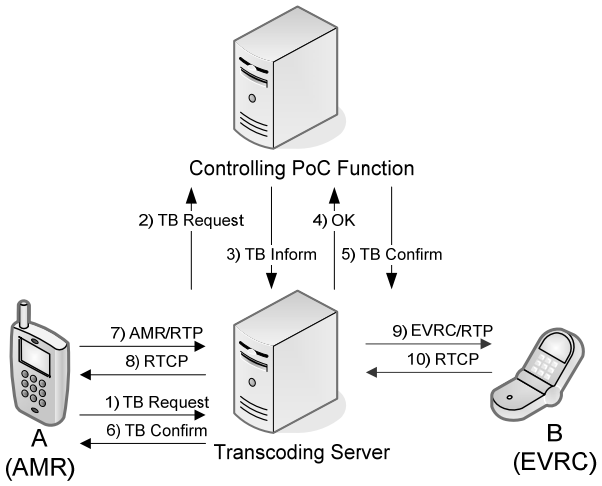

Figure 5: Example of control flow for transcoding centralized at the CPF when a new user has permission to speak.

Note: In this document, we make the TB (Request/Confirm) messages flow between the TS and the CPF for illustration purposes. However, in a real system, we can use an IP switch to route such packets directly to the CPF without wasting TS resources.

3.3. Media streams managed by the CPF

Regarding media streams, the CPF must manage two types of traffic: TBC and the usual media. The first type relates to the requests to speak and the responses between the PoC clients and the CPF. The second type relates to the usual audiovisual media streams, such as AMR over RTP and RTCP packets. Each media stream is assigned a specific port number.

For the media flow, two options are possible:

1. All the media packets arrive at the CPF (see Figure 6). The CPF processes the TBC packets arriving at the TBC Protocol (TBCP) port, while it forwards the usual media streams to the TS. The TS performs transcoding and either returns the result to the CPF (which in turn forwards them to the destination) or sends them directly to the destination.

2. All the media packets arrive at the TS (see Figure 7). The TS forwards the TBC packets arriving at the TBCP port to the CPF, while it transcodes the usual media streams and sends them to their destination. The CPF manages messages arriving from TBCP port and returns the result to the TS (which forwards them to the destination) or sends them directly to the destination.

From a scalability and general performance perspective, option 2 is more efficient, as it minimizes the flow of information arriving at the CPF. The CPF should manage sessions and not have to deal with media packets.

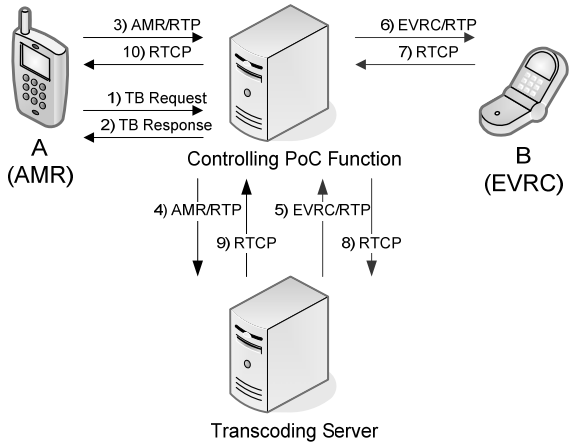

Figure 6: Example of media flow option 1. All media packets arrive at the CPF.

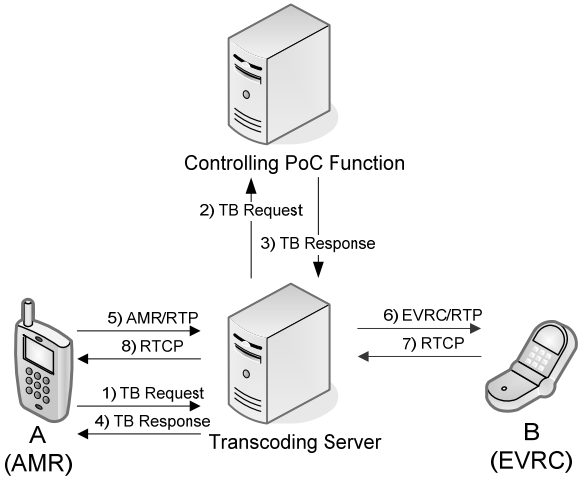

Figure 7: Example of media flow option 2. All media packets arrive at the TS.

4. DISCUSSION

There are several advantages to this solution:

1. The transcoding is transparent for all PoC clients. The solution only requires changes to servers.
2. The solution is compatible with existing PoC specifications.
3. It works for all PoC group session scenarios (1 to 1, 1 to many, 1 to many to 1), ad hoc and pre-arranged, as well as chat group sessions.

4. It is scalable, since the CPFs of many operations can be offloaded by routing the media flow through the TS.
    - The TSs can even be distributed (there can be more than one server).
5. It can be extended to other SIP/SDP-based real-time multi-party sessions.
6. It allows the processing resources required for transcoding to be minimized. For instance, if many destinations require the same transcoded format, transcoding is performed once and the result is delivered to many.
7. It allows the transcoding operation to be customized for each user. For instance, we could select a distinct AMR bitrate for every PoC client.

The main disadvantage of our solution is the added complexity required at the level of the CPF to manage session control operations and media streams.

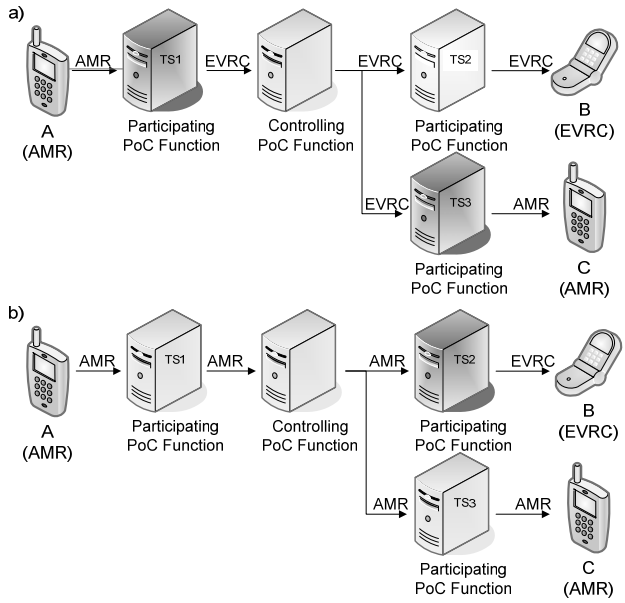

Figure 8: Transcoding performed at different PPFs: a) at the sending and receiving PPFs; b) only at the receiving PPF.

Other transcoding solutions could have been considered. For instance, we could manage and perform the transcoding operations locally at the PPF level. In that case, we must first determine the best location for transcoding, since the transcoding can be performed at the sending PPF or at the receiving PPF. From a speech quality perspective, when multiple users are involved in a session, it is best to perform the transcoding at the receiving user's PPF. This would prevent double transcoding, as illustrated in Figure 8a, where the CPF has decided that EVRC is the codec to be used by all users for the session In Figure 8b, the person who has permission to speak uses the codec format it supports, and the receiving PPF transcodes only if the receiving client does not support that codec format. This is the most reasonable solution. However, this is a sub-optimal one from a computing perspective, as PPFs from various networks (even from the same network) may perform precisely the same transcoding operation on the same media stream for different users. For these reasons, and others that are beyond the scope of this short paper, we believe that centralizing the transcoding operation is the best option.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we have presented a transcoding system for real-time multi-user PoC sessions centralized at the Controlling PoC Function. The solution has several advantages, such as scalability, applicability to all PoC group session scenarios, compatibility with existing PoC specifications, and, most importantly, transparency to existing PoC clients. In future work, we propose to investigate a proxy-based transcoding solution which does not require any change to PoC servers already deployed or to PoC clients.

6. REFERENCES

[1] Open Mobile Alliance, "Push to talk over Cellular (PoC) – Architecture," version 1.0.2, September 2007, OMA-AD-PoC-V1_0_2-20070905-A.

[2] 3GPP TS 26.235 v7.4.0, "Packet switched conversational multimedia applications; Default codecs (Release 7)," March 2008.

[3] 3GPP2 S.R0100-0, "Push-to-Talk over Cellular (PoC) System Requirements," version 1.0, September 2005.

[4] Open Mobile Alliance, "Push to talk over Cellular (PoC) – Architecture," candidate version 2.0, February 2008, OMA-AD-PoC-V2_0-20080226-C.

[5] IETF RFC 3261, "SIP: Session Initiation Protocol," Standards Track, June 2002.

[6] Open Mobile Alliance, "Push to talk over Cellular (PoC) – Control Plane Document," v 1.0.2, September 2007, OMA-TS-PoC_ControlPlane-V1_0_2-20070905-A.

[7] Open Mobile Alliance, "Push to talk over Cellular (PoC) – User Plane," version 1.0.2, September 2007, OMA-TS-PoC_UserPlane-V1_0_2-20070905-A.

[8] 3GPP TS 23.228 v7.11.0, "IP Multimedia Subsystem (IMS); Stage 2 (Release 7)," March 2008.

[9] 3GPP TS 24.229 v7.11.0, "IP Multimedia Call Control based on SIP and SDP; Stage 3 (Release 7)," March 2008.

[10] 3GPP2 X.S0013.2-B, "IP Multimedia Subsystem (IMS); Stage 2," version 1.0, December 2007.

[11] 3GPP2 X.S0013.4-B, "IP Multimedia Call Control Protocol, Based on SIP and SDP stage 3," version 1.0, December 2007.

[12] IETF RFC 4566, "SDP: Session Description Protocol," Standards Track, July 2006.