2E-3

Real-Time Communication Convergence Mechanism within the Network

Junnosuke YAMADA† Alexander BLOTNY‡ Niklas BLUM‡
Michio SHIMOMURA† Thomas MAGEDANZ‡* Naoki UCHIDA†
NTT†, Franunhofer FOKUS‡, TU Berlin*

ABSTRACT The desire to make real-time communication (RTC) with the others in remote places was initially realized by the telephony system. Today the vast development of ICT and network technologies has enabled an anywhere/anytime service availability, e.g., a video conference on mobile devices connected over the Internet. But communication channels are still separated regarding media stream connections and media processing technologies. This research has investigated an architecture in the network to converge, modify and split media streams.

1. Introduction

Wide installations and developments of PSTNs coupled with hard-wired networks have enabled voice RTC between two parties in a distance at anytime. This has been emerged to the mobile telephony system extending to anywhere. It shall be noticed that signaling has been already separated from voice very early in traditional PSTNs.

Apart from these, non-real-time data communication technology has evolved to IP technologies and to the Internet, which has become very popular. They include VoIP technologies such as SIP and RTP.

For further developments of mobile telephony and PSTN, IP technologies and networks have been widely adopted as in NGN/IMS. Furthermore, communication services are nowadays also provided by various non-telco service providers in the Internet such as Skype [1] and based on the emerging WebRTC [2] communication stack affording audio/video RTC even between mobile devices. They employ the Internet only as “bit pipe.” But the network would be a most rational point to be equipped with additional capabilities mostly on media processing (e.g., Speech to Text (STT), Text to Speech (TTS), Automatic Speech Recognition (ASR)) then media stream manipulations (e.g., converge, modify, split) for RTC. It is because that the (mobile) device might be usually poor on processing, difficult to deploy and manage the capabilities, inefficient to process and to handle media streams of RTC.

A single media example for our RTC approach would be NTT docomo’s cloud-based interpretation service [3]. For further developments of RTC, a network architecture to handle multiple media stream and processing capabilities flexibly and smartly is admired and has been tackled. Our proposed architecture in the network enables a multi-media approach to converge, modify and split media streams processed by the capabilities in the network.

† Nippon Telegraph and Telephone Corporation (NTT),
NTT Service Evolution Labs., 1-1, Hikari-no-oka, Yokosuka, Kanagawa 239-0847, Japan
‡ Fraunhofer-Institut für Offene Kommunikationssysteme (FOKUS), Kaiserin-Augusta-Allee 31, 10589 Berlin, Germany
* Technische Universität Berlin (TU Berlin),
Lehrstuhl Architekturen der Vermittlungsknoten (AV),
Franklinstr. 28-29, 10587, Berlin, Germany

2. Architecture Requirements and Design

The architecture was designed from the two aspects, namely, functional components (static) and how they work and are employed (dynamic). The latter means, more precisely, to figure out control patterns of communications with media streams and to define adequate signaling and its interfaces. The signaling shall be self-contained and clearly defined. The following subsections provide underlying architectural aspects of our approach.

2.1. Static Aspects

Static aspects of our architecture are as in the following:

- a flexible mechanism to adopt the capabilities to the users and media streams,
- an advanced mechanism to employ and control multiple media processing capabilities,
- adequate signaling interfaces to control the capabilities and media stream, and
- some sets of signaling to the interfaces according the possible control patters.

The first two will be realized respectively by the functional components called *Controller* and *Capability Adaptor* connected via an interface for clearness and modularity.

The *Controller* manages media stream sessions, users and provides a service registry, data base, state machine and external control interfaces.

The *Capability Adaptor* abstracts individual capabilities and connects to them through capability specific adaptors. The last two will be left to the next subsection.

The whole of the *Controller*, *Capability Adaptor*

Copyright ©2013 Information Processing Society of Japan.
All Rights Reserved.

(including media processing capabilities) with the I/Fs defined later will be called as “RTC Convergence Mechanism” (RTCCM) in the network hereafter.

2.2. Dynamic Aspects
To design signaling, here possible signaling interfaces and patterns should be investigated.
Interfaces: the standard possible interfaces would be on (1) the same plane of media streams, (2) a specific control plane separated from media streams, and (3) a RESTful interface provided for NON-client sides, which are needed, e.g., for controlling video conference concentrated by a capability.
The signaling decoded in media streams is excluded assuming specific technologies and beyond our scope.

Patterns: There are two patterns according to whether or not the RTCCM mediates the media stream.

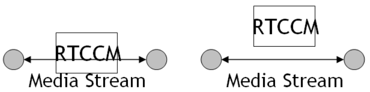

Fig.1 Two Positions of RTCCM to Media Stream

The possible manipulations considering the two positions are:

- to establish and close a media stream in the both position,
- to re-establish or change a media stream with preserving the position,
- to re-establish or change a media stream with changing the position, or
- to control media processing capability under RTCCM when RTCCM mediate a media stream.

2.3. The General Architecture of RTCCM
Compiling the two preceding subsections, the general, simplified architecture below is obtained.

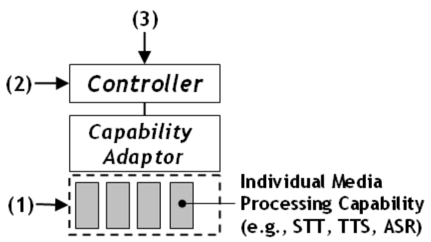

Fig.2 The General Architecture of RTCCM

2.4. Miscellaneous Requirements
It would be very natural to reside both this RTCCM and the capabilities on the cloud connected to the transport IP network for scalability, on-demand resource availability, and robustness.
These additional requirements will be more on software to be applied, which is beyond our current scope.

3. Some Examples
3.1. A Sketchy Use Case
A two-party video conference is held and other participants can be invited from I/F(1) by a command on I/F(3) on RTCCM. Next, some of the three can ask RTC I/F(3) to start STT capability under RTCCM and superimpose its output over media-streams, also create new stream to each parties to show a demo video.

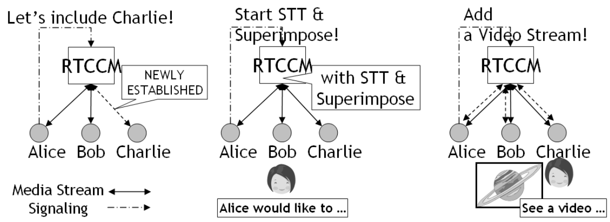

Fig.3 Three Party Video Conference Example

3.2. A Technical Example
Assuming RTCCM is incorporated with IMS, namely, SIP works as signaling with parameters conveyed by SDP. Then, change a stream via I/F(2) on RTCCM not mediating media stream.

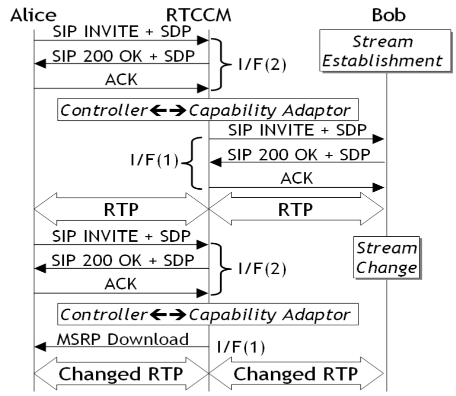

Fig.4 Change Media Stream

4. Further Research
Ongoing further research is performed with the help of a prototype implementation of the environment and investigates the convergence of different communication overlays as IMS and the emerging WebRTC standard. Additionally, the dynamic scalability of the software is investigated, i.e. the dynamic addition of capabilities during runtime and elastic behaviour of individual media capabilities based on resource requirements.

Bibliography
[1] Skype, http://www.skype.org
[2] WebRTC initiative, http://www.webrtc.org
[3] Cloud-based translator phone, http://www.nttdocomo.com/features/mobility36/

Copyright ©2013 Information Processing Society of Japan. All Rights Reserved.