**3GPP TSG-SA WG4 Meeting #127S4-240253rev1**

**Sophia-Antipolis, France, 29 January - 2 February 2024**

**Source: Interdigital Finland Oy, Nokia Corporation**

**Title: [AI4MEDIA] pCR on AI/ML model distribution procedures**

**Spec: 3GPP TR 26.927 v0.5.0**

**Agenda item: 9.6**

**Document for: Discussion and agreement**

**1. Introduction**

This contribution proposes:

1. Update the Basic workflow for AI/ML clause §5.2.1.2-1 of the TR 26.927 v0.5.0
   1. Details the selection of a model including evaluation of multiple criterions, for example based on model precision, UE capabilities (e.g. the available processing power, the battery level), the inference latency and the network latency.
   2. Update the Basic workflow for AI/ML clause §5.2.1.2-1 to include basic model update steps.
2. Update basic workflow for progressive/adaptive model delivery update §5.2.1.2-2 of PD v1.01 and move the procedure to TR 26.927 v0.5.0
   1. Discussion and change of the name "progressive download” to “adaptive download”. The top-level specification 26.501 details delivery methods and procedures i) clause 5.2.2 “Progressive download of on-demand content” where the media segments of a media content are downloaded to the client device and played back as it downloads and ii) DASH (Dynamic Adaptive Streaming over HTTP) clause 5.2.3 where the client selects, downloads and plays different media segments to adjust the quality. For AIML model “progressive” delivery, the client downloads a first model subsets/segments and infers it, then downloads, and infers additional model subsets/segments to achieve a better model precision. The client UE somehow “adaptively” selects and downloads model subsets/parts based on the conditions at the time of downloading using of any of delivery method above 5.2.2 and 5.2.3.
   2. Update example of adaptive model compositions. 1) additive composition such as addition of bits for a bit-incremental model (e.g. quantized model 8, 16, 32 bits) or addition of neurons for a pruned model. 2) consecutive composition by appending model data to previously received model data. (e.g. model with different part including early exits).
   3. Generalized the progressive/adaptive update procedure to include the selection of a Full model or an Adaptive model. For example, the UE may first evaluate the latency to download either a full model or an adaptive model with a low level of precision as an initial download.

**2. Reason for Changes**

The reasons to change:

1. We propose to update 5.2.1.2-1 to add detailed example on the model selection and to include a basic update to the existing workflow.
2. We propose to update text of basic workflow for progressive model delivery:
   1. change “progressive “delivery to “adaptive” delivery to better describe the delivery type and to avoid further confusion with the TS 26.601 progressive download. An adaptive model may be delivered using the progressive download method similar to 26.501 or any other delivery method.
   2. Give example on different adaptive model compositions.
   3. Include the selection of either a Full model or an adaptive model in the adaptive scenario. Both cases are relevant for the selection of an adaptive model.

**3. Proposal**

We propose to update the clause §5.2.1 of the TR 26.927 v0.5.0 and update §5.2.2 of the permanent document PD v1.01 with the proposed changes.

\* \* \* Begin of Changes \* \* \* \*

#### 5.2.2.2 Basic workflows

Figure 5.2.2-2 shows a basic workflow for AI/ML model delivery with inference in the UE. Steps for the procedures shown are described below.

Une image contenant texte, reçu, ligne, diagramme

Description générée automatiquement

Figure 5.2.2-2: Basic workflow for AI/ML model delivery with inference in the UE

During the initialization and establishment step, it is assumed that information related to the required features and detailed configurations are exchanged and negotiated between the network and UE. Information may include those related to UE device and network capabilities, AI/ML service information (e.g. service requirements, AI/ML model descriptions), and delivery methods. Such information may be used for the selection of a suitable AI/ML model for the service.

1. The UE Application and Network Application communicate to trigger AI model delivery, using the information from the initialization and establishment step.

2. An AI model is selected between the UE Application and Network Application.

3. The Network Application identifies the selected AI model in the AI model Repository/Provider.

4. The AI Model Access Function establishes an AI model delivery session with the AI Model Delivery Function.

5. The AI Model Access Function receives the AI model.

6. The AI Model Access Function passes the AI/ML model to the AI model Inference Engine in the UE.

7. The Data Source passes media data to the AI model Inference Engine.

8. The AI Model Inference Engine performs AI inferencing.

9. The AI Model Inference Engine passes the inference output result to the UE Data Destination for consumption.

Adaptive model delivery refers to a model delivery paradigm wherein a smaller size but lower precision model is delivered to a UE first to speed up the inference at the UE and to improve QoE. Subsequent model updates are delivered to the UE and the model at the UE is updated to a higher precision. In this context, an adaptive model refers to a model which can be used for inference as it is by the UE, but subsequent updates can be applied to it to improve its accuracy. The update may be applied in different ways to compose the model, depending on how the low precision model is built. For example, 1) additive composition such as addition of bits for a bit-incremental model (e.g. quantized model 8, 16, 32 bits) or addition of neurons for a pruned model. 2) consecutive composition by appending model data to the previously received model data. (e.g. model with different subsets including early exits).

Figure 5.2.1.2-2 and text below shows a basic workflow for adaptive model delivery update. Steps for the procedures shown are described below.



**Figure 5.2.1.2-2: Basic workflow for adaptive model delivery update**

* 1. During the initialization and establishment step , The UE Application and Network Application communicate to establish adaptive model delivery. The UE Application may receive Service Access information to learn about available services and configurations, including available AI models, precisions and possible updates. This information may be in a 3GPP URI of/or model manifest file(s). The model manifest file contains size, complexity information etc. of the different versions. The available model list may comprise full models (as for 5.2.1.2-1), or adaptive models.
  2. An adaptive model is selected by the UE Application, based on, e.g. model size and currently available network capacity.
  3. The UE application requests the adaptive model of selected precision from the Network Application
  4. The Network Application identifies the selected AI model in the AI model Repository/Provider.

Adaptive AI model delivery session loop

* 1. The AI Model Access Function establishes an AI model delivery session with the AI Model Delivery Function.
  2. The AI Model Access Function receives the AI model of the precision requested by the UE.
  3. The AI Model Access Function passes the AI/ML model to the AI model Inference Engine in the UE.
  4. The Data Source passes data to the AI model Inference Engine, AI Model Inference Engine performs AI inferencing,
  5. The *AI Model Inference Engine* performs AI inferencing.
  6. AI Model Inference Engine passes the inference output result to the UE Data Destination for consumption.

Mode delivery update.

* 1. The UE application triggers a model precision update for updating the AI model to a higher precision.

AI Model delivery session is reused or established according to step 5-10. These steps may be repeated depending upon number of precision levels and corresponding model updates.

* 1. The update is applied to the low precision model.

The inference loop of step 9 continues.

\* \* \* End of Changes \* \* \* \*