**Source: Interdigital New York**

**Title: [FS\_AI4Media] pCR on AI related user plane data**

**Spec: 3GPP TR 26.927 v0.9.0**

**Agenda item: 9.6**

**Document for: Agreement**

1. **Introduction**

The contribution S4aV240065 presented in the ad-hoc telco demonstrated that the size of intermediate data cannot be determined before the inference time for some split point configurations evaluated on object detection models. The same apply for LLM model designed with variable-length input data. The conclusion is that the size of the tensor data should be transmitted as metadata in the user plane with the intermediate data.

Like the size of intermediate data, other metadata parameters can only be determined at inference time and must be transmitted with intermediate data in the user plane.

This contribution adds a section describing AI-related user plane data for split operations with the following changes:

* The first change is to add a definition of user-plane metadata to the clause 5.3.1. “AI data components” and a new clause 5.3.X describing the pipeline of split inferencing tasks.
* The second change is to add a new “user-plane data for split inferencing” section to clause 6 “Data components for AI/ML-based media services “. The section first and then details the different reasons why user-plane metadata information needs to be carried with intermediate data. An illustrated example, taken from a ssd\_resnet object detection inferencing, is described.

1. **Reason for Change**

Lack of section presenting AI-related user plane data for split operations.

1. **Proposal**

It is proposed to agree the following changes to 3GPP TR 26.927 v0.9.0

\* \* \* First Change \* \* \* \*

## 5.3 Architecture for AI data delivery

### 5.3.1 AI data components

AI-related user plane data includes:

- AI model data, including data describing the topology/structure of the AI model, data related to the data nodes of the model, i.e. tensors, and other data which may be dependent on the format used for the AI/ML model.

- Intermediate data, defined as the output data from the inference process of an AI/ML model that is not considered the final inference result (depending on the service and output layer of the split AI model, certain intermediate data may have media characteristics, or even be media data). Intermediate data is typically required to be delivered to a second device or entity, as the input to a subsequent second split inference.

- Inference output data, which is the data corresponding to the output result of the final AI inference process for the service. Depending on the nature of the AI data inferencing for the given AI data service, this inference output data may include: labels for identifying recognition like tasks from media, actual media data such as video and/or audio, or perhaps XR related data such as 3D models.

- Inference input data, corresponds to all inputs feeding the AI inference. In case of a split inference, input data feeds the first inference starting the inference at the input of the trained model. For AI for media use-cases, input is media data (image, video, audio, etc.)

- Training input data, corresponds to all inputs feeding the AI training process on a device for federated learning. Such data is typically created on, or exists in UE devices.

- Training results data, which is the data corresponding to the output result of the AI training process. Such data is typically delivered by a UE device to a federated learning entity which aggregates data from multiple UE devices to update and train a model

* User-plane metadata, corresponds to contextual and additional information to the data payload being transmitted.

### 5.3.X Split inference tasks pipeline

The following Figure 5.3.X-1 describes the split inference tasks pipeline comprising media capture, a pre-processing task to adapt input media data to inference input data, split inference of head or tail parts exchanging intermediate data and metadata, post-processing to transform inference output data to a media data representation (text/image/video) for media consumption.



Figure 5.3.X-1 Split inference tasks pipeline

Figure 5.3.X-2 describes a breakdown of the pipeline between two endpoints where the first endpoint contains the media data source and initiates inference. Endpoints can be a UE or an Application Server (AS).



Figure 5.3.X-2 Split inference pipeline breakdown between endpoints

\* \* \* end first Change \* \* \* \*

\* \* \* second Change \* \* \* \*

## 6 Data components for AI/ML-based media services

## 6.X AI-related user plane data

Section §5.3.1 introduces the different AI-related user plane data type namely, AI model data, intermediate data, inference input data, inference output data, training input data, training results data and user-plane metadata.

### 6.X.1 User-plane data for split inferencing

### 6.X.1.2 User-plane metadata with intermediate data

Metadata for split operations described in clause 6.6 can be negotiated between two endpoints at the negotiation stage before the inference loop starts. However, metadata may also need to be carried out during the inference stage with intermediate data. The reasons why the identified user-plane metadata information need to be transmitted with intermediate data payload are listed below:

* A split operation on a model comprising multiple branches produces several tensors. Therefore, at the inference stage, the intermediate data payload contains multiple tensors. Metadata information must be transmitted to identify each tensor data in the intermediate data payload at the receiver endpoint:
  + Tensor identifiers
  + Tensor type
* Some models can be designed to infer media data with variable size. Splitting these models can produce intermediate tensor data of varying shapes. In this case, the size of the intermediate tensors can only be calculated at inference stage. Tensor data size characteristics need to be transmitted with the tensor data:
  + Tensor shape
  + Tensor type
* When a compression algorithm is reselected, for example to modify the compression ratio of the intermediate data or to increase the prediction quality, information on the (re)selected compression algorithm and related tensor information may need to be transmitted with tensor data:
  + Tensor identifier
  + Tensor shape
  + Tensor type
  + Tensor compression algorithm profile identifier

The following table summarizes the user-plane metadata that may need to be carried with intermediate data payload as described above.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Metadata category** | **Metadata type** | | **Definition** | **Examples** |
| **User-plane metadata** | Tensor list | | List of Tensors that composed the intermediate data |  |
|  | Tensor identifier | A unique identifier for the tensor. The identifier may be a name, an index of a tensor list or table, a combination thereof, a hash value. | Tensor1  10 |
|  | Tensor shape | Tensor shape is a tuple of positive integers, where the size of the tuple represents the dimension of the tensor, and each value represents the size in each dimension. | [1,64,64,64]. |
|  | Tensor data type | The data type of each intermediate data tensor | Float32, int32 |
|  | Tensor compression algorithm profile identifier | Identifies the selected compression algorithm profile | FCM high 5.1, FCM main 5.3, FCM 6.4  NNC xxx 5.7.9, NNC yyy 5.8, NNC yyy 6.4 |
| **Intermediate data payload** |  | Tensor data |  | [] |

The following table shows an example of user-plane data, including metadata applied to an object detection inference of the *ssd\_resnet* model for the different scenarios:

* Multibranch split at node 30 where each tensor must be identified among the intermediate data tensors (Relu\_output\_0, Conv\_output\_0),
* Tensor shape changing at runtime, with a variable image dimension feeding the second part of the dplit inference (e.g. [1,256,75,75])
* Reselection of a compression profile settings applied for the same or different split point (e.g. Conv\_output\_0 with a new quantized parameter qp=-14).

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **data category** | **data type** | | | **Data example** |
| Intermediate data tensor | Intermediate data tensor element list | | |  |
|  |  | Tensor identifier | Relu\_output\_0 |
|  |  | Tensor shape | [1,256,75,75] |
|  |  | Tensor data type | float32 |
|  |  | Tensor compression profile Identifier | nnc.xx (with qp=-14) |
|  |  | Tensor data | [] |
|  |  | Tensor identifier | Conv\_output\_0 |
|  |  | Tensor shape | [1,512,38,38] |
|  |  | Tensor data type | float32 |
|  |  | Tensor compression profile Identifier | nnc.xx with qp=-14 |
|  |  | Tensor data | [] |

\* \* \* End of second change \* \* \* \*