April 2023

AIDED REMOTE RENDERING IN A 5G-AS-A-SERVICE MOBILE NETWORK ENVIRONMENT

Vimal Srivastava

Sri Gundavelli

AIDED REMOTE RENDERING IN A 5G-AS-A-SERVICE MOBILE NETWORK ENVIRONMENT

AUTHORS:
Vimal Srivastava
Sri Gundavelli

ABSTRACT

Artificial Intelligence (AI)/Machine Learning (ML)-based mobile applications are becoming increasingly computation-intensive, memory-consuming, and power-consuming. In addition, end devices usually have stringent energy consumption, compute, and memory limitations for running a complete offline AI/ML inference on-board. Many AI/ML applications currently offload inference processing from mobile devices to internet data centers (IDC). Techniques presented herein provide for discovering the closest offload server so that UE can offload some of the rending work to the offload server.

DETAILED DESCRIPTION

Image and video are the biggest data on today's Internet. Videos account for over 70% of daily Internet traffic. Convolutional Neural Network (CNN) models have been widely used for image/video recognition tasks (e.g., image classification, image segmentation, object localization and detection, face authentication, action recognition, enhanced photography, virtual reality (VR)/ augmented reality (AR), video games, etc.) on mobile devices. Meanwhile, CNN model inference requires an intensive computation and storage resource.

The AI/ML-based mobile applications are increasingly computation-intensive, memory-consuming, and power-consuming. End devices usually have resource limitations for running a complete offline AI/ML inference on-board. Many AI/ML applications (e.g., image recognition) currently offload inference processing from mobile devices to IDCs. For example, photos taken on a smartphone are often processed in a cloud AI/ML server before being shown to the user who took them.

Model splitting is the most significant feature for AI inference. The number of terminal computing layers and an amount of data transmitted correspond to different model splitting points. In other words, when a user device or user equipment (UE) has low computation capacity (e.g., due to low battery), an application running on the UE can change the splitting point to let the UE calculate fewer layers while increasing the data rate in the Uu interface for transmitting a higher load of intermediate data to the network.

Figure 1, below, illustrates an example in which a side link is used to realize the proximity-based work task offloading.

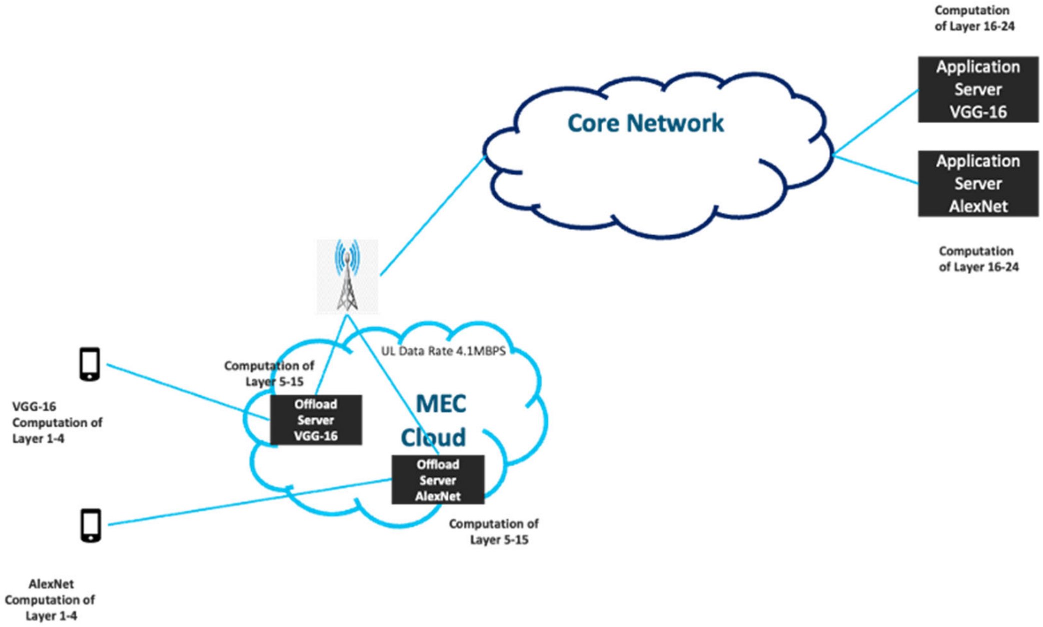
Computation of Layer 16-24
Application Server VGG-16
Core Network
Application Server AlexNet
Computation of Layer 16-24
UL Data Rate 4.1MBPS
Computation of Layer 5-15
Offload Server VGG-16
MEC Cloud
VGG-16 Computation of Layer 1-4
Offload Server AlexNet
Computation of Layer 5-15
AlexNet Computation of Layer 1-4

Figure 1: Example Network Diagram

By using a side link to realize the proximity-based work task offloading, the data rate on the Uu interface does not need to be increased while the original UE's computation load is offloaded. This leads to battery savings in the UE. Techniques presented herein provide for facilitating the UE discovering the closest offload server that uses the same rendering model as the UE to allow the UE to offload rendering work to the offload server.

Techniques presented herein provide for configuring a gNodeB with offload servers and information related to the offload servers (e.g., an offload server's L-2 address, model, split, rendering details, etc.). Consider that the gNodeB may be configured statically (e.g., Operations, Administration and Maintenance (OAM)) or dynamically (e.g., via the Application Function (AF)/Access and Mobility Function (AMF)) by an operator. Consider that the gNodeB broadcasts the information in a System Information Block (SIB). A UE looking to offload rending may use this information to select an offload server. The UE may also use a Domain Name System (DNS) query (e.g., based on location, rending model, etc.) to find an offload server to offload rendering.

Figure 2, below, illustrates a communication sequence diagram that shows steps for discovering the closest offload server that uses the same rendering model as a UE that is attempting to offload processing.

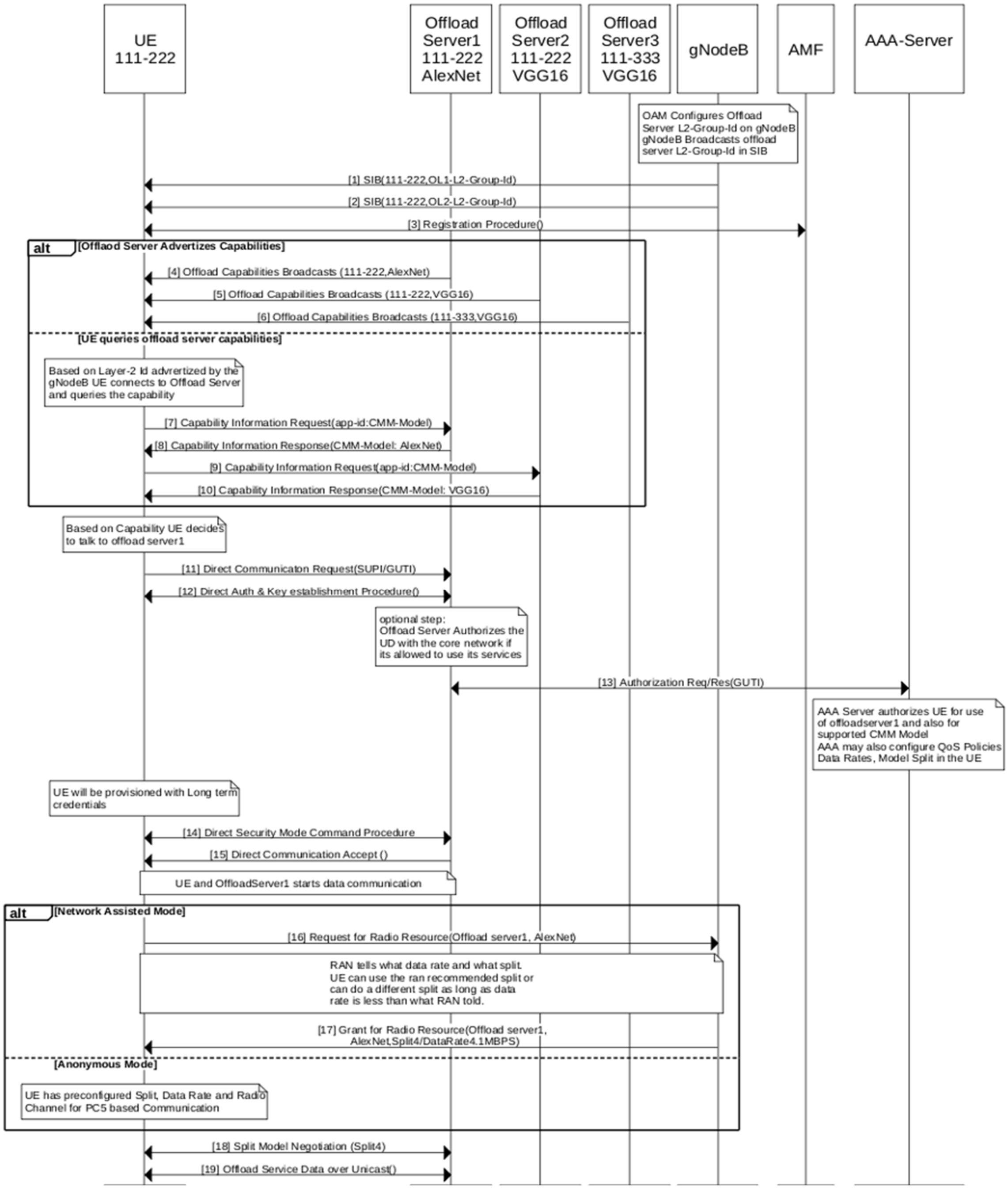
UE
111-222
Offload
Server1
111-222
AlexNet
Offload
Server2
111-222
VGG16
Offload
Server3
111-333
VGG16
gNodeB
AMF
AAA-Server
OAM Configures Offload
Server L2-Group-Id on gNodeB
gNodeB Broadcasts offload
server L2-Group-Id in SIB
[1] SIB(111-222,OL1-L2-Group-Id)
[2] SIB(111-222,OL2-L2-Group-Id)
[3] Registration Procedure()
alt
[Offlaod Server Advertizes Capabilities]
[4] Offload Capabilities Broadcasts (111-222,AlexNet)
[5] Offload Capabilities Broadcasts (111-222,VGG16)
[6] Offload Capabilities Broadcasts (111-333,VGG16)
[UE queries offload server capabilities]
Based on Layer-2 Id advrertized by the
gNodeB UE connects to Offload Server
and queries the capability
[7] Capability Information Request(app-id:CMM-Model)
[8] Capability Information Response(CMM-Model: AlexNet)
[9] Capability Information Request(app-id:CMM-Model)
[10] Capability Information Response(CMM-Model: VGG16)
Based on Capability UE decides
to talk to offload server1
[11] Direct Communicaton Request(SUPI/GUTI)
[12] Direct Auth & Key establishment Procedure()
optional step:
Offload Server Authorizes the
UD with the core network if
its allowed to use its services
[13] Authorization Req/Res(GUTI)
AAA Server authorizes UE for use
of offloadserver1 and also for
supported CMM Model
AAA may also configure QoS Policies
Data Rates, Model Split in the UE
UE will be provisioned with Long term
credentials
[14] Direct Security Mode Command Procedure
[15] Direct Communication Accept ()
UE and OffloadServer1 starts data communication
alt
[Network Assisted Mode]
[16] Request for Radio Resource(Offload server1, AlexNet)
RAN tells what data rate and what split.
UE can use the ran recommended split or
can do a different split as long as data
rate is less than what RAN told.
[17] Grant for Radio Resource(Offload server1,
AlexNet,Split4/DataRate4.1MBPS)
[Anonymous Mode]
UE has preconfigured Split, Data Rate and Radio
Channel for PC5 based Communication
[18] Split Model Negotiation (Split4)
[19] Offload Service Data over Unicast()

Figure 2: Example Communication Sequence Diagram

In a first step, consider that an OAM configures the gNodeB with a list of offload servers that are available in a cell associated with the gNodeB. In a second step, consider that the gNodeB broadcasts the list of offload servers available in the cell for a Public Land Mobile Network (PLMN) in a SIB. In a third step, consider that the UE receives the list of offload servers for its PLMN in the SIB.

In a fourth step, consider that each offload server advertises its supported models (e.g., CNN models) and the computational split. In a fifth step, consider that the UE selects an offload server. Consider that UE selects an offload server by comparing a UE-supported model and split with the offload server's advertised values. The UE may form a DNS query to find the offload server's address. The DNS query may take a PLMN identifier, Tracking Area Identity (TAI), cell identifier, computation model, and split as inputs. The UE may query the core network to get the offload server's Fully Qualified Doman Name (FQDN) based on the UE's location and computational requirements.

As a sixth step, consider that, after the offload server selection is made, the UE establishes PC5 interfaces (anonymous or network assisted). As a seventh step, consider that the UE is configured with a long term security key for PC5-based security establishment. As an eighth step, consider that the UE starts using the desired model and offloading computation to the offload server.

In summary, as provided by the techniques described herein, a UE may discover a closest offload server that uses a rendering model supported by the UE. The UE may offload rendering work to the offload server to conserve resources at the UE.