April 2023

OFFLOAD SERVER SELECTION BASED ON USER EQUIPMENT ROUTE SELECTION POLICY

Sri Gundavelli

Vimal Srivastava

Ravi Kiran Guntupalli

OFFLOAD SERVER SELECTION BASED ON USER EQUIPMENT ROUTE SELECTION POLICY

AUTHORS:
Sri Gundavelli
Vimal Srivastava
Ravi Kiran Guntupalli

ABSTRACT

Many applications offload inference processing from the mobile devices running the applications to Internet datacenters. By using a side link to realize proximity-based work task offloading, the data rate on the Uu interface does not need to be increased while the original device's computation load is offloaded. This leads to battery savings in the device. Techniques described herein outline efficient ways for a device to discover an offload server for offloading processing.

DETAILED DESCRIPTION

Image data and video data are the biggest data on today's Internet. Videos account for over 70% of daily Internet traffic. Convolutional Neural Network (CNN) models (e.g., AlexNet) 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) on mobile devices. Meanwhile, CNN model inference requires an intensive computation and storage resource.

Artificial intelligence (AI)/ machine learning (ML)-based mobile applications are increasingly computation-intensive, memory-consuming, and power-consuming. 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 (e.g., image recognition) offload the inference processing from mobile devices to Internet datacenters (IDC). For example, photos taken by 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 the amount of data transmission correspond to different model splitting points. In other words, when a user equipment (UE) has low computation capacity (e.g., due to low battery), the application can change the splitting point to let a UE calculate fewer layers while increasing the data rate in the Uu interface for transmitting a higher load of intermediate data to a network.

Figure 1, below, is a diagram illustrating an offload server in a consumer use case.

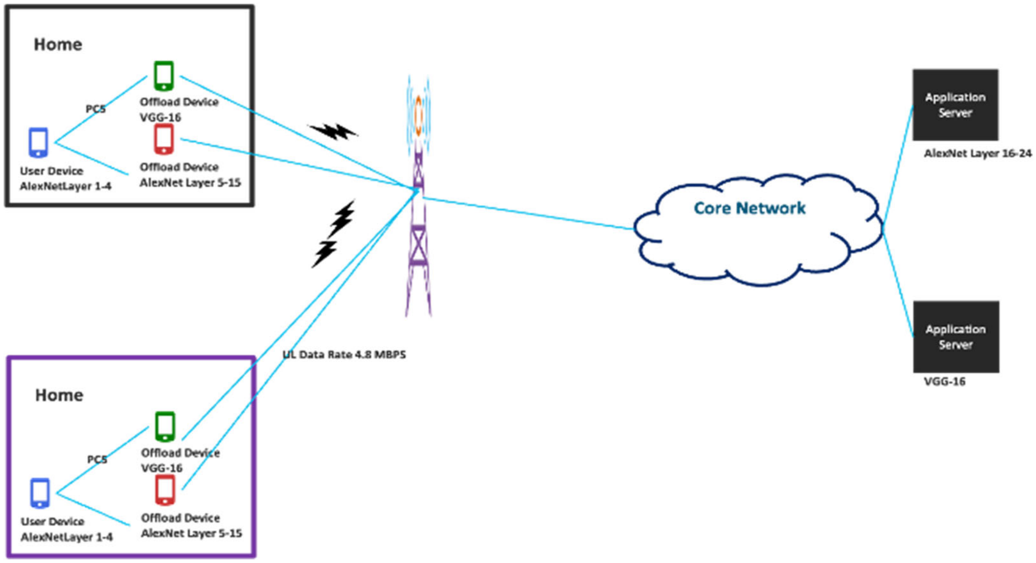

Figure 1: Diagram of an Example Offload Server in a Consumer Use Case

As illustrated in Figure 1, 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 described herein outline ways for a UE to discover an offload server for offloading processing.

One solution for discovering an offload server involves extending the UE policy and including OSSP (Offload Server Selection Policies) along with Access Network Discover and Selection Policies (ANDSP), UE Route Selection Policies (URSP), and Vehicle-to-Everything Policies (V2XP). OSSP policies will include information about available offload servers the UE can use for offloading computation.

Figure 2, below, is a message sequence diagram illustrating two cases for identifying available offload servers. In the first case, the offload server is owned by the operator. In the second case, the offload server is owned by the user.

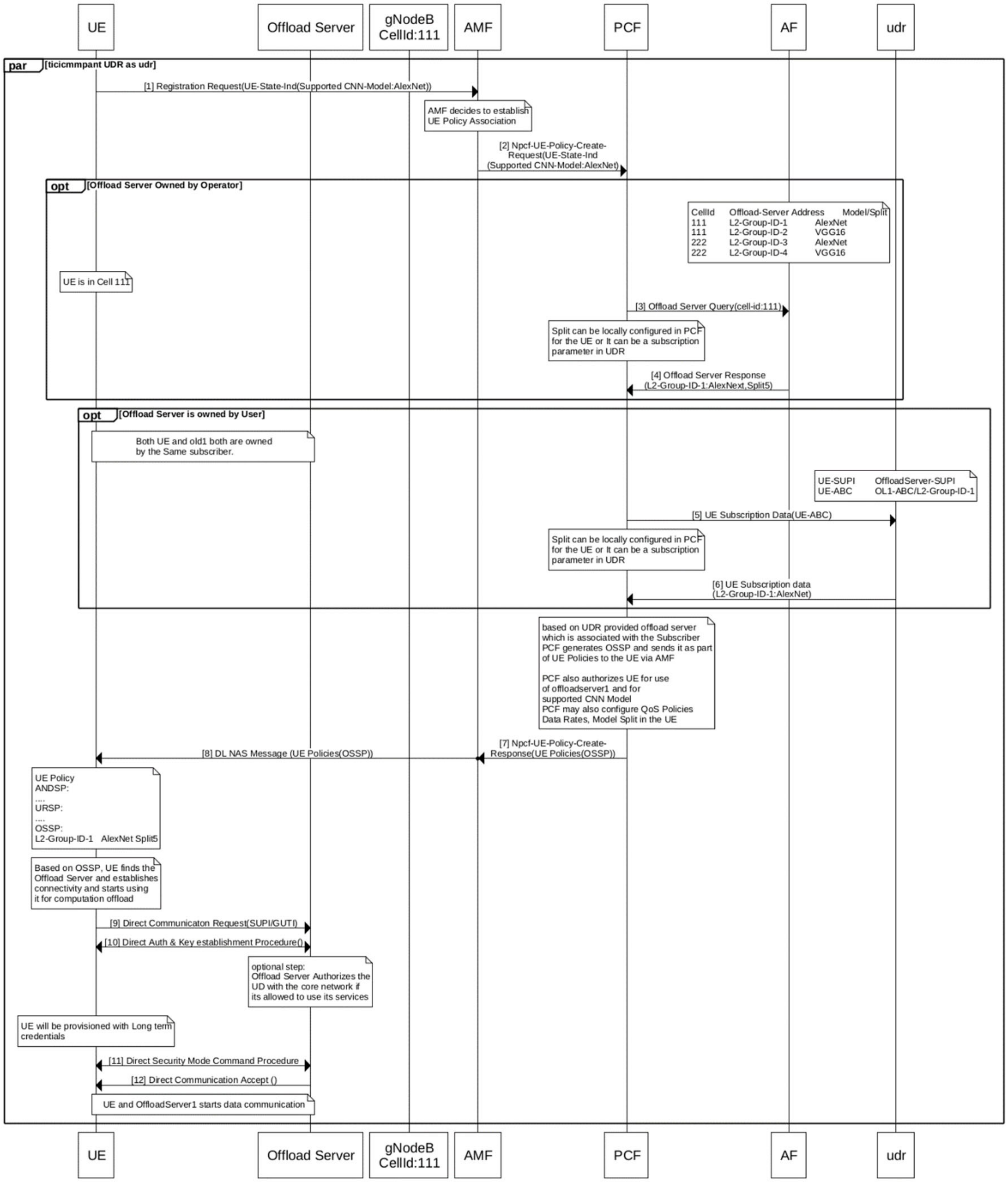
UE
Offload Server
gNodeB CellId:111
AMF
PCF
AF
udr
par [ticicmmpant UDR as udr]
[1] Registration Request(UE-State-Ind(Supported CNN-Model:AlexNet))
AMF decides to establish UE Policy Association
[2] Npcf-UE-Policy-Create-Request(UE-State-Ind (Supported CNN-Model:AlexNet)
opt [Offload Server Owned by Operator]
CellId Offload-Server Address Model/Split
111 L2-Group-ID-1 AlexNet
111 L2-Group-ID-2 VGG16
222 L2-Group-ID-3 AlexNet
222 L2-Group-ID-4 VGG16
UE is in Cell 111
[3] Offload Server Query(cell-id:111)
Split can be locally configured in PCF for the UE or It can be a subscription parameter in UDR
[4] Offload Server Response (L2-Group-ID-1:AlexNext,Split5)
opt [Offload Server is owned by User]
Both UE and old1 both are owned by the Same subscriber.
UE-SUPI OffloadServer-SUPI
UE-ABC OL1-ABC/L2-Group-ID-1
[5] UE Subscription Data(UE-ABC)
Split can be locally configured in PCF for the UE or It can be a subscription parameter in UDR
[6] UE Subscription data (L2-Group-ID-1:AlexNet)
based on UDR provided offload server which is associated with the Subscriber PCF generates OSSP and sends it as part of UE Policies to the UE via AMF
PCF also authorizes UE for use of offloadserver1 and for supported CNN Model
PCF may also configure QoS Policies Data Rates, Model Split in the UE
[7] Npcf-UE-Policy-Create-Response(UE Policies(OSSP))
[8] DL NAS Message (UE Policies(OSSP))
UE Policy
ANDSP:
....
URSP:
....
OSSP:
L2-Group-ID-1 AlexNet Split5
Based on OSSP, UE finds the Offload Server and establishes connectivity and starts using it for computation offload
[9] Direct Communicaton Request(SUPI/GUTI)
[10] Direct Auth & Key establishment Procedure()
optional step: Offload Server Authorizes the UD with the core network if its allowed to use its services
UE will be provisioned with Long term credentials
[11] Direct Security Mode Command Procedure
[12] Direct Communication Accept ()
UE and OffloadServer1 starts data communication
UE
Offload Server
gNodeB CellId:111
AMF
PCF
AF
udr

Figure 2: Message Sequence Diagram of Example Cases for Identifying Offload Servers

In the first case illustrated in Figure 2, the offload server is owned by the operator. Techniques described herein outline a four-step solution for identifying an offload server that is owned by the operator. In a first step, the UE registers with the network and indicates its supported CNN model. In a second step, the network determines the available offload server based on the UE location/tracking area identifier (TAI)/Cell-ID. In a third step, the network provides OSSP information to the UE. In a fourth step, the UE finds the closest OSSP and starts using it for offloading computation.

In the second case, the offload server is owned by the user. Techniques described herein outline a four-step solution for identifying an offload server that is owned by the user. In a first step, the UE registers with the network and indicates its supported CNN model. In a second step, the Policy and Control Function (PCF) queries the User Data Repository (UDR) and identifies an associated offload server for the UE. In a third step, the PCF constructs the OSSPs for the UE based on available information. In a fourth step, the UE finds the closest OSSP and starts using it for offloading computation.

According to techniques described herein, the OSSP Information may include the Layer 2 Group ID of the Offload Server, PC5 mode of connection (Network Assisted, Anonymous), and Supported Models/Computation Split.

According to techniques described herein, an additional solution for discovering an offload server involves sending offload server information to the UE via a Protocol Configuration Options (PCO)/enhanced PCO information element (protocol option). The Application Function provides this information to the Session Management Function (SMF) and the SMF includes the offload server information.

According to techniques described herein, in an additional solution for discovering an offload server, the UE may have configured a Fully Qualified Domain Name (FQDN) for an offload server it can use to get the address of the IP server.

In summary, techniques described herein provide a UE with offload server information by extending the UE policies.