Title: NDN Compute Simulator (ndnCSim)

Target: NDN research community

Benefits: In-network computation and specifically edge cloud computing has attracted extensive attention in the recent past from both academia and Industry. The common goal of these technologies is to enable the networking nodes to perform computations. Such networking nodes include 1) edge computing node, 2) the resource-full routers between an edge node and cloud node and 3) the cloud node itself. Various NDN-based in-network computation solutions have been proposed by the researchers, indicating the widespread interest of the NDN research community in edge cloud computing and in-network computations. However, unfortunately, the default implementation of the ndnSIM simulator does not provide the basic realization of in-network computations, leaving the NDN researchers with the only option of implementing the computation mechanism from scratch. To save the researchers time and to provide them the basic implementation of in-network computation, this hackathon project aims to complete the **following task**.

1. Enable the consumer node to set the computation-based naming scheme.
2. Enable the intermediate nodes to perform computations based on the received compute request in the Interest packet. To mimic the computation behavior, we will add different computation models. Moreover, we will also add the functionality to monitor the resource consumption of the intermediate nodes. For example, by default, the intermediate node resource utilization will be 0%, however, as soon the computation requests arrive, the resource utilization will be increased to let say 10% depending on nature (type) of the task. Thus, if 100% resource utilization is met, the intermediate node offloads the request to the next node. All of the above-mentioned implementations will be incorporated into the NFD codebase.
3. Develop NDN compute-based producer application in an apps folder. Similar to the intermediate node, the producer node application also monitors resource utilization. However, different from the intermediate node, the producer node computes the application adds the compute request in the waiting pool if the resources are fully occupied, and performs the computation after the resource availability.

The NDN research community will be greatly benefited from the above-mentioned implementation, as by utilizing ndnCSIM the researcher can leverage the basic implementation of in-network computations in vanilla-ndnSIM.

Tools: C++, vanilla-ndnSIM

Outcome: Demo of the ndnCSIM by running a custom example

Team Size: 4