SD-Core 1.2 Release

Ajay Lotan Thakur
ajay@opennetworking.org

October 25, 2022
Agenda

- Release 1.2 Overview
- SD-Core Introduction
- SD-Core 1.2 Features
- Q & A
SD-Core 1.2 Release Overview

- SD-Core 1.2 Released on Oct 11, 2022
- **Highlight** - Move towards cloud native 5G network functions
- SD-Core 1.2 [Release blog](#)
- SD-Core 1.2 [Release Notes](#)
- Helm Charts can be found on [Charts Repository](#). Helm Charts code is tagged & branch created
- SD-Core Document [Website](#) has SDCore-1.2 branch
- SD-Core NF Code repository code is not tagged but image versions has git commit id included in them, so if required we can create branch/tag from commit hash
- Thank You all Community Members!
  - Intel, Infosys, CPQD, Purdue University, GS Lab
  - There are more individuals contributing with individual Contributor Licencse Agreement (CLA)
SD-Core Support Recap

- 4G/5G/5G-NSA Core Network Support
  - 4G supports - Release 13
  - 5G supports – Release 15.3
- SD-Core is fully containerized, and helm charts provided to deploy SD-Core on K8s
- CUPS compliant Control Plane, User Plane
- 2 UPF implementations available – BESS-UPF, P4-UPF
  - You can connect any other UPF to control Plane
  - BESS UPF is available in AF_PACKET mode, DPDK mode
- Configuration APIs to create/update/delete Network Slice
- Subscribers can be provisioned through simapp APIs or through configmap updates
- 3 Level QoS support – Slice level, UE level and application level
  - In progress - QoS scheduling support
- Application Filtering support
- Access control support
Goals:

- **Goal is to have Multiple Instances of all Network Functions**
  - Currently we achieved multiple instances of AMF, SMF

**3gpp NRF features**

- Keep alive
- Caching NRF results. Could not make to release, code review in progress

**3gpp SET features**

- Multiple Network Functions of same type share the User Context and User state is not lost if some members of the SET are removed/restarted

**MongoDB started in cluster mode**

- MongDb streaming feature is used
- Timeout on the documents is used
- Multiple keys are added in hash table
NRF Keep Alive Support

- Feature is default disabled. Enabled by changing override values for NRF
- Keep Alive Support in NRF
  - Configuring Heartbeat Timer Value
  - Handling Heartbeat Messages from the NFs
  - Deletion of NFProfile from DB on Heartbeat timer expire
  - Discover returns multiple profiles
- Keep Alive Support in other NFs
  - Sending NFProfile update message to NRF on Heartbeat Timer expire
  - Send Deregistration message to NRF (graceful exit)
- Future Release
  - NRF cache and NRF notification support shall come in upcoming release
AMF Changes

- New K8s Pod SctpLb
- Accept and manage gNodeB connections
- GRPC communication between SctpLb and Amf Service
- Handles Amf Instance Down/Up Notifications
- Distributed Resource Sharing Module (DRSM) used to share resource IDs among AMFs
- SctpLb is disabled by Default and can be enabled through helm chart override values
SCTP Load Balancer

- Tracks AMF instances in the deployment
- SCTPLb just reads SCTP socket message. No ngap message decoded.
- Round-Robin Distribution of Sctp Messages over grpc channel to AMF
- Redirect Support for forwarding Sctp Messages to a particular Amf Instance
- If instance is detected to be down, then remove the AMF from Pool
- Future release
  - Decode NGAP message and forward message to correct AMF instance
  - Multiple SctpLb support
    - Will help in scaling number of gNBs connected to core.
  - NRF api integration in SctpLb to discover as custom NF
AMF DB support

- DB Read when context is not found in memory
- DB create/update on success of following procedures
  - Registration Procedure
  - Deregistration Procedure
  - Service Request Procedure
  - PDU Session Setup Procedure
  - PDU Session Release Procedure
  - AN Release Procedure
  - GUTI Registration Procedure
  - Network Initiated Deregistration Procedure
  - Context Replacement: Registration on Registration
- DB Delete on Completion of following procedures
  - Deregistration Procedure
  - Network Initiated Deregistration Procedure
- REST API support to fetch Active UEs Context from DB.
- REST API to purge Subscriber from AMF
SMF Changes

- Introduced UPF adapter as PFCP endpoint from SMF side
- UPF adapter talks to one or more UPF
- UPF-Adapter acts like Mux-Demux for PFCP messages between multiple SMF instances and UPF.
- The UPF-Adapter intercepts and modifies certain fields before sending PFCP msg to actual UPF.
- Similar handling is done for the response from UPF towards SMF.
- PDU Session preservation in Database
- Two or more SMF instance supported. Scale up/down of SMF instance supported
SMF Changes

- **PDU Session preservation in Database**
  - SMF shall preserve PDU Session in Database after every procedure to maintain latest updates. This shall ensure stateless support to operate on PDU session at any SMF instance.

- **UPF support for handling multiple SMF instances association**
  - UPF-Adapter introduced to handle multiple SMF PFCP associations with same Node-Id towards UPF.

- **NRF Keep-Alive support. Send periodic NRF profile update**

- **Unique ID/IP-Address Management across all the Instances**
  - Distributed Resource Assignment handling using DRSM
  - UPF supported UE address allocation
  - With support of DRSM module, during the recovery of any SMF instance, other NFs shall claim resources of that instance and shall become new owner of the resources(e.g., SEID/TEID Pools)

- **AMF shall forward ongoing procedure to alternate SMF if procedures times out/fails with current SMF instance**
  - Retransmission of message to alternate SMF
  - NRF notification will reduce the signalling delays during pod restart
gNBSim Changes

- Support for Custom Profile
  - Iterate over procedures as per configured steps
  - Delay between procedures possible
  - StepTrigger to execute procedures one by one. Step profile can be enabled. Step trigger can be given through API. [Example here](#).
  - Execute subscribers in parallel to simulate 100+ calls

```yaml
customProfiles:
customProfiles1:
  profileType: custom # profile type
  profileName: custom1 # uniquely identifies a profile within application
  enable: false # Set true to execute the profile, false otherwise.
  execInParallel: false # run all subscribers in parallel
  stepTrigger: true # wait for trigger to move to next step
  gnbName: gnb1 # gNB to be used for this profile
  startImsi: 200930100007487
  ueCount: 5
  defaultAs: "192.168.250.1" # default icmp pkt destination
  opc: "981464c7c62eb6e036234984ad0bcf" # [example]
  key: "512225021433e723a5dd523fc145fc0" # [example]
  sequenceNumber: "16f3b3f70fc2" # [example]
  plmnId: # Public Land Mobile Network ID, <PLMN ID> = <MCC><MNC>
  mcc: 208 # Mobile Country Code (3 digits string, digit: 0-9)
  mnc: 93 # Mobile Network Code (2 or 3 digits string, digit: 0-9)
startIteration: iteration1
iterations:
  #at most 7 actions
  - name: "iteration1"
    1: "REGISTRATION-PROCEDURE 5"
    2: "PDU-SESSION-ESTABLISHMENT-PROCEDURE 5"
    3: "USER-DATA-PACKET-GENERATION-PROCEDURE 10"
    next: "iteration2"
  - name: "iteration2"
    1: "AN-RELEASE-PROCEDURE 100"
    2: "UE-TRIGGERED-SERVICE-REQUEST-PROCEDURE 10"
    repeat: 5
    next: "iteration3"
  - name: "iteration3"
    1: "UE-INITIATED-DEREGISTRATION-PROCEDURE 10"
    # "repeat": 0 #default value 0, i.e. execute once
    # "next": "quit" #default value quit, i.e. no further iteration to run
```

gNBSim Changes

- Regular/default profiles
  - Procedure list is fixed per profile
  - Execute Subscribers In Parallel to simulate 100+ calls
  - Step profile can be enabled. Step trigger can be given through API. Example here.
  - Custom Profile and Regular profile
  - Delay between procedures can not be provided

- Add new calls in exiting profile
  - Works when step profile is enabled

- REST Api to trigger new profile execution
- Dnn, slice Information can be configured
- Future release
  - gNB handovers, gNB deployment as container, error indication, gtpu echo handling, negative testing, external interface to send uplink traffic

```yaml
profiles: # profile information
- profileType: register # profile type
  profileName: profile1 # uniquely identifies a profile within application
  enable: false # Set true to execute the profile, false otherwise.
  gnbName: gnb1 # gNB to be used for this profile
  startImsi: 208930100007487
  ucCount: 5
  defaultAs: "192.168.250.1" #default icmp pkt destination
  opc: "981d464c7c52eb6e5036234984ad0bcf"
  key: "5122250214c33e723a5dd523fc145fc0"
  sequenceNumber: "16f3b3f70fc2"
  dnn: "internet"
  sNssai:
    sst: 1 # Slice/Service Type (uinteger, range: 0-255)
    sd: 010203 # Slice Differentiator (3 bytes hex string, range: 000000-FFFFFF)
  execInParallel: false #run all subscribers within profile in parallel
  plmnId: # Public Land Mobile Network ID, <PLMN ID> = <MCC>-<MNC>
  mcc: 208 # Mobile Country Code (3 digits string, digit: 0-9)
  mnc: 93 # Mobile Network Code (2 or 3 digits string, digit: 0-9)
```
Documentation Improvements

- Updated Configuration Guide
- UPF deployment
- QoS configuration
- Release Notes for 1.2
- Application Filtering Rules
- Default Aether in a Box (AiaB) override file has newly added comments to explain configuration
- Design Section added to capture new design aspects
- Next Release features on doc website
Future Enhancements

• Details of next release targeted features available at doc website
• Documentation for UPF deployment
  – Standalone Control + edge
  – Distributed Edge
• Supporting metrics from multiple NFs and creating dashboard
  – Making dashboard available for AIAB for 5G
• Cloud Native UPF
• gNBSim feature extensions
• Spin up/down NF instances based on metrics such as CPU utilization, memory utilization, or custom metrics
• NRF cache and NRF notification support
• Multiple ScpLb Instance Support
  – NGAP Decode to forward messages to a particular Amf Instance
• Multiple UPF adapter support
  – UPF initiated message handling
• Most of the above Features will be address in SD-Core 1.3 Release, scheduled to be delivered in Jan 2023
How to contribute

• Queries can be sent on slack channel
  – Channels #sdcore-dev #aether-dev

• Identifying value add features and send the proposal for review

• Aether Users
  – Let us know documentation issues
  – Raise pull request on code, document, helm charts

• Developer
  – Raise pull request for code changes

• Quality Assurance
  – Test our released code and let us know issues found

• Join our Aether Community calls. Next meeting October 27th | 8:00am Pacific
  https://opennetworking.org/events/aether-community-meeting-kaejet/

• Previous meeting recordings available here
Thank You!