RAN Simulation for ONAP / SMO Use Cases

N. K. Shankaranarayanan\textsuperscript{1}, John Keeney\textsuperscript{2}, Alex Stancu\textsuperscript{3}, Nikita Trivedi\textsuperscript{1}
Martin Skorupski\textsuperscript{3}, Ahila Pandaram\textsuperscript{4}, Krishna Moorthy\textsuperscript{1}
(\textsuperscript{1}-STL, \textsuperscript{2}-Ericsson, \textsuperscript{3}-highstreet technologies, \textsuperscript{4}-Wipro)

Jan 13, 2022

@twitterhandle
Outline

• Introduction
• RAN Simulation for ONAP/SMO use cases
• ONAP Use Cases
• ONAP and OSC RAN Simulators
• Cellular radio simulation (ns3)
• Future work
• Discussion
RAN Simulation for end-to-end use cases

- Perspective: ONAP/OSC end-to-end use cases involving ONAP and SMO components
- Simulation of RAN includes:
  - Support for SMO to RAN Interfaces
  - Abstraction of RAN components/functionality as per use case
- Natural to have different types of RAN Simulators
- Overall objectives for ONAP/OSC community:
  - Reusable solutions - efficiency and ease of future integration
  - Avoid duplication and divergence
RAN Simulation: Use Case viewpoint

Use Case functions implemented:
• ONAP Layer / SMO
• Models, microservices, applications, data handling, control loops, policy

RAN functions simulated:
• Interfaces to/from SMO
• Functional logic and data flow in RAN components
• Cellular radio network performance

RAN Simulation areas

Interface simulation
• Focus on O-RAN interface
• e.g., O1 netconf server, VES client
• Configuration of RAN (O1, M-plane)
• Data from network (O1, M-plane)
• Guidance to xApps (A1)
• Cloud config/data (O2)

Functional simulation
• Logic, data flow for Use Case
• xApp/E2 processing
• CU/DU/RU function

Cellular network simulation
• Radio channel conditions, UE mobility,
  user plane performance (throughput, latency, loss)

Test-Stubs / Simulators / Emulators

- Test-Stub
  - Very basic small skeletal placeholder, (usually) stateless, with (usually) hardcoded responses to a very small subset of possible inputs/contexts.
  - Often used to verify connectivity, non-functional aspects, aid test coverage, or as an “early-stage simulator”

- Simulator
  - Mimics a subset of behavior of a simulated function/interface/system with correct data/information models
  - Behaves or operates like “the real thing” when provided with a set of controlled inputs
    - Realism/Fidelity is a key differentiator!

- Emulator
  - Indistinguishable / exactly like “the real thing” for a wide range of usage scenarios
  - Often used for validation/testing late in the development process

- Simulator/Emulators can only be an approximation of “The Real Thing”
  - Usefulness of a simulator is entirely subjective – depending on the consumer’s custom usecase/requirements
    - Best ones are customizable/extensible/configurable/model-driven
  - Trade-off: advantages vs realism/fidelity

- Advantages of Simulator/Emulator
  - More convenient, accessible, extensible, repeatable, cheaper, safer, timely, more/less/appropriately realistic

Pointers to definitive descriptions would be very welcome!
ONAP Use Cases
(using ONAP RAN-Sim Simulator):

• ONAP SON Use Case
• ONAP Slicing Use Case
ONAP SON Use Case: Rel 3 to Rel 9

- **SON ↔ Control Loop (CL)**
- **Companies can use framework to add proprietary SON solutions, including optimization algorithms, etc.**

- **OOF-SON use case has built a foundation for ONAP/O-RAN integration**
- **Radio network uses common netconf/yang model**

**Data flows**
- SDN-R to RAN: netconf-based configuration
- RAN to DCAE: VES format for FM alarms, PM KPI, CM Notification

**Control loop**
- Data Collection, & Analysis (DCAE)
- Optimization (OOF)
- Co-ordination, Decisions (Policy)
- Action (SDN-R)
- RIC (non-RT)
- O-RAN Radio Network (RAN-Sim)

**Data, CM, FM, PM**

**Config ack**

**O1**
- CM notify
ONAP SON Use Case – Release 10+

- SON ↔ Control Loop (CL)
- Companies can use framework to add proprietary SON solutions, including optimization algorithms, etc.

- OOF-SON use case has built a foundation for ONAP/O-RAN integration
- Radio network uses common netconf/yang model

Data flows
- SDN-R to RAN: (O1) netconf-based configuration, (A1) Policy/guidance to xApp in Near-RT RIC
- RAN to DCAE: (O1) VES format for FM alarms, PM KPI, CM Notification
ONAP SON Use Case – Control Loop with O1 & A1 (Rel 10/11)

Steps 3a,4a,4b,4c: CM data received from RAN, update of state in CPS DB
Steps 3d,4d: FM/PM data received from RAN, processing in SON MS
Steps 2,4e,4f, 5: SON MS analyzes need for optimization
Steps 6,7,7b,8: Optimization using OOF
Steps 9,10,11b,11c: Automated action to make change in RAN (O1 and A1)
Steps 11,12,13: Action status

Jakarta Release:
1. New VES Message format to align with O1
2. Complete CM-Notify carryover work
3. Include A1-based action (stretch goal, may be planning only)
E2E Network Slicing: Architecture & Interfaces

Notes:
1. This interface is needed only for deployment scenario 1 (see later slides).
2. Several aspects are also aligned to ETSI ZSM.
3. Not yet implemented

Transport Network Configurations

3rd Party RAN NSSMF
RAN NSSMF
TN NSSMF
Core NSSMF

3rd Party Core NSSMF

Transport Slice Subnet 1
Transport Slice Subnet 2
Transport Slice Subnet 3

Front Haul
Mid Haul
Back Haul

RAN Slice Subnet
Core Slice Subnet

RAN
DU
CU
Core

3rd Party RAN NSSMF

Align with 3GPP
Align with TSCi (IETF)
Align with 3GPP
Align with TMF
Align with 3GPP

RAN NSSMF
TN NSSMF
Core NSSMF

RAN Slice Subnet
Core Slice Subnet

Transport Network Configurations

On Univ.

Front Haul
Mid Haul
Back Haul

RAN
DU
CU
Core

Transport Slice Subnet 1
Transport Slice Subnet 2
Transport Slice Subnet 3

RAN Slice Subnet
Core Slice Subnet

Transport Network Configurations

On Univ.

Front Haul
Mid Haul
Back Haul

RAN
DU
CU
Core

Transport Slice Subnet 1
Transport Slice Subnet 2
Transport Slice Subnet 3

RAN Slice Subnet
Core Slice Subnet

Transport Network Configurations

On Univ.
**E2E Network Slicing – Closed Loop**

- **Config DB**
  - RAN config data
  - Closed loop triggered

- **Policy**
  - Closed loop triggered

- **DCAE**
  - Slice Analysis
  - MS
  - PM-Mapper

- **SO**
  - SO as an Actor

- **SDN-R**
  - Domain controller applies the new configuration to RAN

- **RAN Simulator**

**Data flow from RAN n/w to DCAE-Slice Analysis MS**

- **VES Consumer**
  - VES Collector
  - Data File Collector
  - PM Mapper

- **DMoAP**
  - Message Router
  - Data Router

Ref. ONAP wiki
The PM data collected from RAN in Step 1 is DL/UL PRB used for data traffic.

The configuration update determined by Slice Analysis MS and triggering Policy in Step 4 is slice specific throughput guidance for Near-RT coverage area (i.e., at Near-RT RIC level).

Step 8 is over O1 in Honolulu, it will be over A1 in Istanbul (using A1 adaptor).

- **Deferred**

**Notes:**
1. DFC and VES Collector are not shown in the flow but are used.
2. Initial configuration may also be over A1 (based on Slice Profile decomposition) in the future.
RAN-Simulator (RAN-Sim)
Approach for RAN Simulator (RANSIm) – Dublin - FM

- **GUI**
  - Provide Control to User in UI to select a cell and modify neighbor list to simulate collision/confusion

- **Controller**
  - Send message to corresponding Netconf Server about the collision/confusion

- **Netconf Server**
  - Send configChangeNotification to SDNR
  - **Send FM/alarm message to VES Collector for collision/confusion. Thus the control loop starts**

---

- **GUI**
  - Show the modified PCI values

- **Controller**
  - When the modified RAN configuration received, same is updated into DB and informed to UI

- **Netconf Server**
  - When the SDNR provides the modified RAN configuration, same is forwarded to Controller
Approach for RAN Simulator (RANSim) – Dublin - PM

- **GUI**
  - Provide Control to User in UI to select a cell and generate PM data

- **Controller**
  - Send PM message to corresponding Netconf Server with KPI values. Initially send good values and after few minutes send bad values for selected cell.

- **Netconf Server**
  - When the SDNR provides the modified RAN configuration, same is forwarded to Controller

- **GUI**
  - Show the ANR updates as dotted lines.

- **Controller**
  - When the modified RAN configuration received, same is updated into DB and informed to UI

- **Netconf Server**
  - Send PM message to VES Collector with given KPI values. Thus the control loop starts.
Enhancement to include xApp and A1

SDN-R
(One or more instances)

RANSIM controller

Near-RT-RIC(Honeycomb)

xapp

xapp

O-CU (Honeycomb)

E2

E2

O-DU (Honeycomb)

O1

O1

O1

O1

RAN-Sim
Problem: RAN-Sim Upgrade is Needed

• All SON Use Case planned work has impact on RAN-Sim
• RAN-Sim uses Honeycomb netconf server
• Honeycomb project (https://wiki.fd.io/view/Honeycomb) has been archived, and future support is an issue
  – There is a new project Hc2VPP (https://wiki.fd.io/view/Hc2vpp) which uses Honeycomb
• Other parts of RAN-Sim provide great value for ONAP and OSC Use Cases
  – Controller, Database, GUI for Usecase logic
  – VES message generation for PM, FM, CM
• Need to develop plan for:
  – Replacement of Honeycomb in RAN-Sim
  – Adding simulated Near-RT RIC and xApp in RAN-Sim for Use Cases
  – Harmonization of RAN-Sim with OSC projects (sim, o1)
O1 simulator in OSC sim
Simulated devices in Simulated RAN are netconf (netopeer) devices representing CU/DU O1 instances with CU/DU yang models.

NTS (Network Topology Simulator) Manager uses netconf to connect to devices using yang model of simulation entities.

NTS also has NB netconf interface for external entity (e.g. SDN-R) to control simulation.

Datastore in NTS provided by netopeer.
O-RAN-SC O1 Simulator architecture

Simulated NF (O-CU/O-DU/O-RU)

NETCONF Server framework – Netopeer2
Comparing RAN-Sim and OSC sim

RAN-Sim

- RAN-Sim was developed for ONAP Use case
- Has use case specific logic (e.g. analyze cell relations and generate fault message)
- Netconf devices are mounted manually
- VES message details are generated in Controller as per use case logic

OSC sim

- OSC sim developed as O1 building block
- NTS can be used by SDN-R to spawn O1 (CU/DU) instances. SDNR GUI can be used to configure NTS
- FM VES messages are pre-configured in O1 sim
- NTS uses netconf to each O1 sim netopeer device, and REST to the docker
ONAP NF Simulator

- xNF with O1 interface
- Netconf server
- VES Client
- File-based PM
- Config Attribute Change

A1 Functions in ONAP / OSC

Ref: https://wiki.onap.org/pages/viewpage.action?pageId=84672221

- A1-related message can be sent from Policy to SDN-R/CCSDK
- Current Control Loop architecture would use DMaaP CL message
- A1 Policy Management Service can connect to A1-termination in the Near-RT RIC with or without an A1 Adapter
- In the SON Use Case context, the A1-termination in Near-RT RIC would be part of the simulated RAN (RAN-Sim) with usecase logic in the simulated/abstracted xApp which results in events from CU/DU over O1
OSC A1 (Near-RT RIC) Simulator

Stateful A1 test stub for “South-end” of A1 Interface

- Used to create multiple stateful A1 providers (simulated near-rt-rics)
  - Supports A1-Policy and A1-EnrichmentInformation
- Does not include and app-specific behavior – just what is defined in A1 Application Protocol specs
- Implemented as a lightweight Python application
- Swagger-based northbound interface, so easy to change the A1 profile exposed (e.g. A1 version, A1 Policy Types, A1-E1 consumers, etc)
  - All A1-AP versions supported as configurable profiles

Source: https://gerrit.o-ran-sc.org/r/admin/repos/sim/a1-interface
Docs: https://docs.o-ran-sc.org/projects/o-ran-sc-sim-a1-interface
Wiki: https://wiki.o-ran-sc.org/display/RICNR
Cellular Network Simulator: ns-3

- ns3 provides models of how packet data networks perform, and provides a simulation engine for users to conduct simulation experiments
- Allows the prototyping of algorithms for
  - Radio Resource Management
  - Self Organized Networks
  - Inter-cell Interference Coordination
- PHY layer abstraction
- Realistic Data Plane Protocol stack model
- Simplified Control Plane model
- Simplified EPC- One MME and one SGW and PGW.
- KPIs availability at different levels:
  - Channel- SINR maps, pathloss matrices
  - PHY- TB tx/rx traces, RSRP/RSRQ traces
  - MAC-UL/DL Scheduling traces
  - RLC and PDCP – Time averaged PDU Tx/Rx stats
  - IP and application stats- Flow monitor, PCAP traces

Ref: https://www.nsnam.org/docs/models/html/lte.html
ns3 Design Choices

- Data Plane Protocol stack model
  - Realistic RLC, PDCP, S1-U, X2-U
  - Allow proper interaction with IP networking
  - Allow end-to-end QoE evaluations
- Control Plane Protocol model:
  - Realistic RRC model
  - Simplified S1-C, X2-C and S11 models
  - Realistic S1-AP, X2-C, S5
- Simplified EPC
  - One MME and one SGW
  - SGW and PGW in the same node (no S5/S8 interface)
- Focus on connected mode
  - RRC connected, EMM Registered, ECM connected
E2E flows: RAN Interfaces + Cell Network

Interface Simulation
- O1 netconf server
- O1 VES Client
- A1 Termination

Functional Simulation
- Use Case specific logic
- xApp processing
- CU/DU processing

Cellular Network Simulation
- Radio channel
- UE Mobility/Handovers
- Data plane performance

Common Framework?
Observations

- Different use cases need different abstractions for efficient implementation -> expect RAN Simulator variations
- There is opportunity to create common building blocks
  - Yang models for RAN
  - Netconf server/client building block
  - VES stndDefined domain and schemas for O1
  - Network topology models
  - Common formats/lists for PM KPI, FM fault
  - Integration of ML methodologies
  - Abstractions of RAN functionality
RAN Simulator Enhancements

- Modification of ONAP RAN-Sim netconf server
- Extension of OSC sim for use case logic
- Simulation of Near-RT RIC / xApp to complement SMO/rApp implementation
- Simulation support for RU endpoint and configuration
- Simulation support for O2 interface and cloud functionality
- Transport network effects
- Enhancement of cellular network simulation
- Common framework for combining simulators
RAN Simulation areas

Interface simulation
- Focus on O-RAN interface
- E.g. O1 netconf server, VES client
- Configuration of RAN (O1, M-plane)
- Data from network (O1, M-plane)
- Guidance to xApps (A1)
- Cloud config/data (O2)

Functional simulation
- Logic, data flow for Use Case
- xApp/E2 processing
- CU/DU/RU function

Cellular network simulation
- Radio channel conditions, UE mobility, user plane performance (throughput, latency, loss)

Backup
O-RAN Control Loops

- **Non-RT Loop**
  - Time scale: ~ secs/mins
  - Direct config of CU/DU
  - Policy Guidance, Coordination

- **Near-RT Loop**
  - Near-Real-Time (~100ms)
  - Based on E2 service models

- **SON examples**
  - Non-RT: Changes based on operational state, averaged behavior
  - Near-RT: Changes based on radio channel, mobility