P4-Based In-Band Telemetry for OSM-Orchestrated 4G/5G Testbeds

Thomas Dreibholz, dreibh@simula.no
Andrés Felipe Ocampo, andres@simula.no
Mah-Rukh Fida, mahrukhfida@simula.no

OSM #12 Ecosystem Day
December 1, 2021

This project has received funding from the EU's Horizon 2020 research and innovation programme under grant agreement No 815279.

Table of Contents

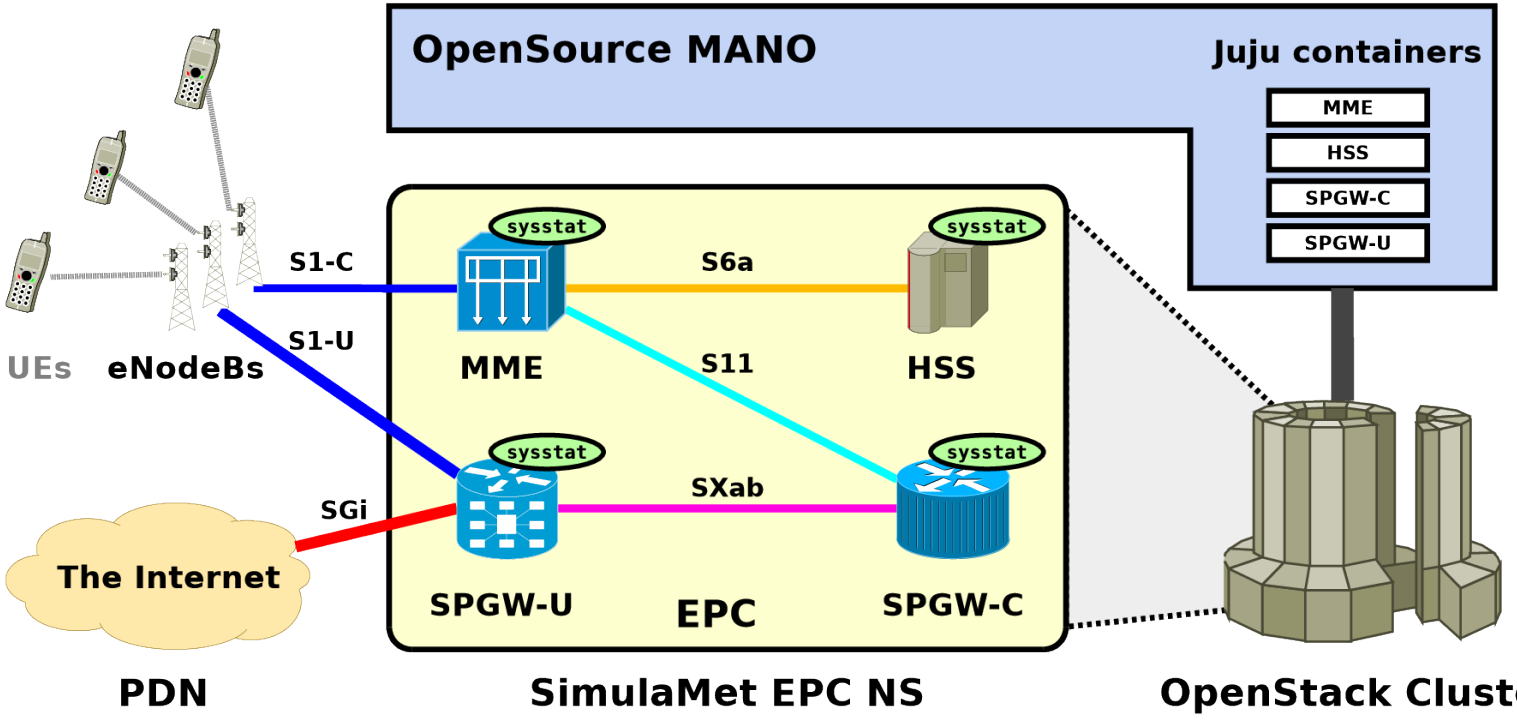
Motivation

- 4G/5G networks need monitoring:
  Detect problems, evaluate performance, etc.
- Basic performance: metrics in OSM, via Juju and NFVI
  Useful, but not very fine-granular
- In-Band Telemetry:
  - Fine-granular monitoring at packet level
  - Possibility to add custom information into packets
    (additional headers, etc.) and process them elsewhere
- Our approach:
  - Usage of Programming Protocol-independent Packet Processors (P4)
  - Custom P4 software switches as VDUs in our testbed

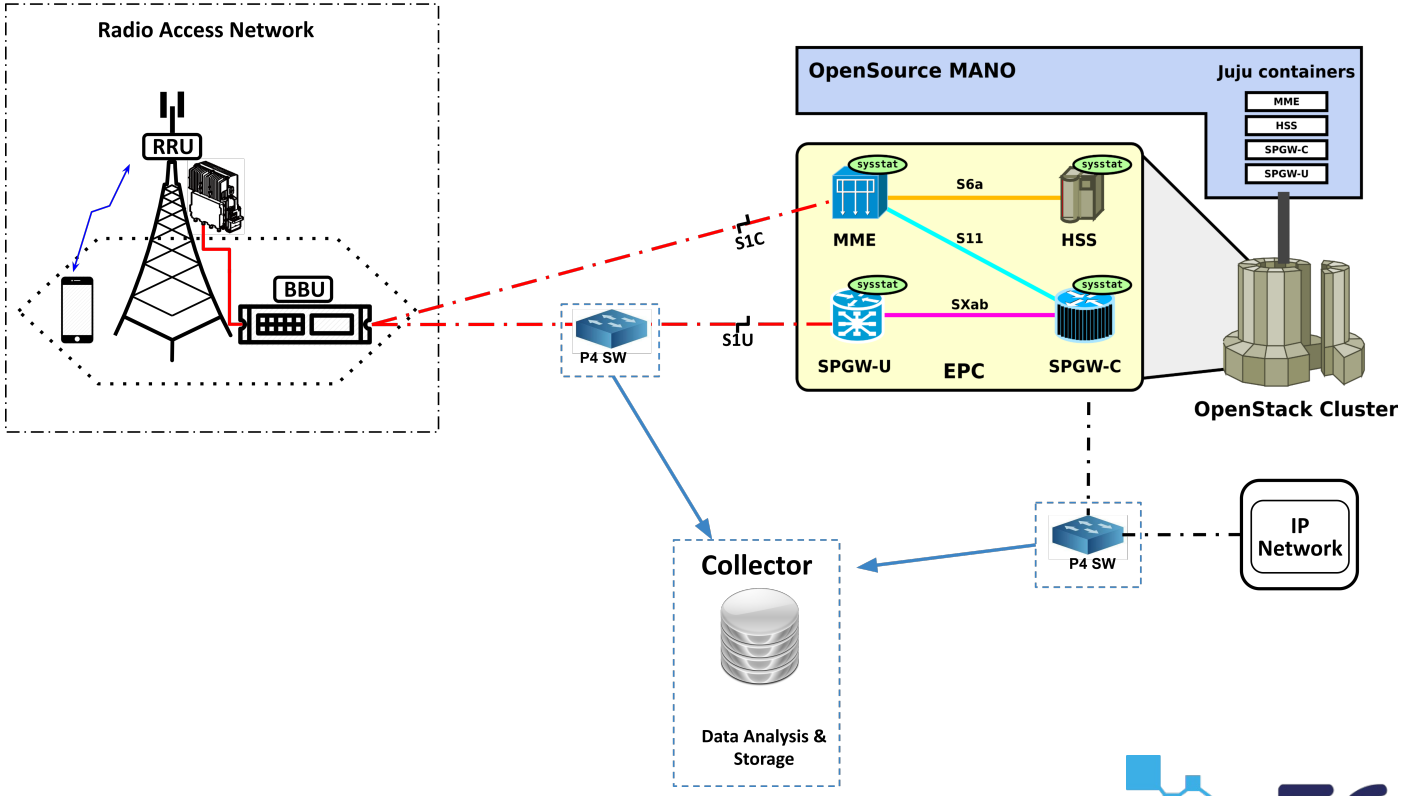
OpenAirInterface (OAI)

- OpenAirInterface (OAI):
  - Open Source software for EPC and eNodeB (i.e. packet core and base stations)
  - Details: https://www.openairinterface.org
  - 4G LTE available, 5G under development
  - Ongoing work, with many different Git branches
- Idea:
  - Manage OAI setups in OSM (at least, the EPC part)
    - Automatic setup and deployment
  - Easy to add additional features (e.g. Mobile Edge Computing components)
  - Open Source, of course! → https://github.com/simula/5gvinni-oai-ns

Basic Testbed Setup



Telemetry Testbed



P4 Program at Switch at S1U Interface

CLONE PACKET AT THE EGRESS, AND
STORE THE NEEDED METADATA INFORMANTION, TO BE USED FOR THE CLONE PACKET, IN REGISTERS

```
if(standard_metadata.instance_type==0){
         reg_packet_length.write(0, standard_metadata.packet_length);
         reg_deq_qdepth.write(0,standard_metadata.deq_qdepth);
         reg_deq_timedelta.write(0,standard_metadata.deq_timedelta);
        clone(CloneType.E2E, 100);
        }
```

DERIVE TELEMETRY DATA AT EGRESS

```
// RUN MIN COUNT SKETCH, TO COUNT PACKETS OF THE FLOW
compute_flowid();
compute_index();
increment_count();
compute_mincount(meta.count1, meta.count2, meta.count3);

//CHECK IF PACKET IS PART OF BURST
meta.marker=0;
is_heavy_hitter();
```

P4 Program at Switch at S1U Interface

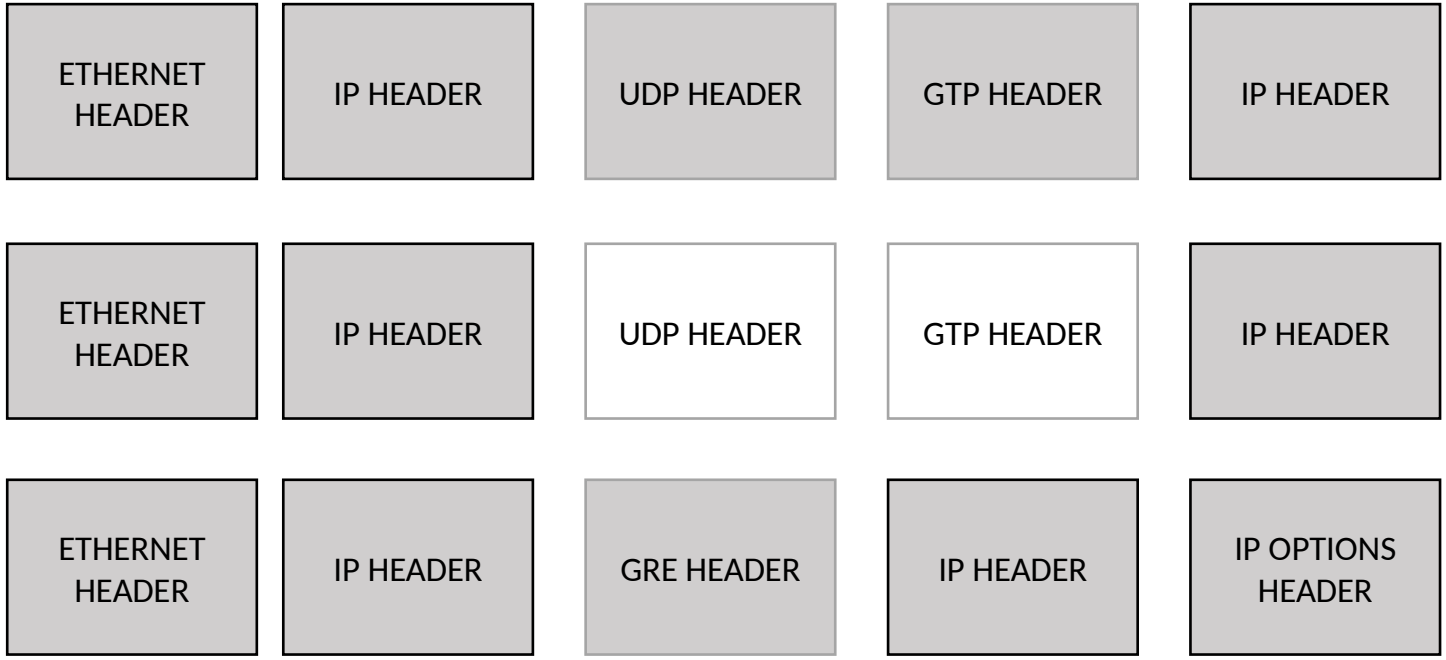
ADD INT TELEMETRY USING IP OPTION HEADER, TO THE CLONE PACKET

```
// SET IP OPTIONS VALID
hdr.ipv4_option.setValid();
hdr.gre.setValid();

//Set UDP, GTP INVALID
hdr.gtp.setInvalid();
hdr.udp_outer.setInvalid();

hdr.ipv4_option.copyFlag = 1;
hdr.ipv4_option.optClass = 0;
hdr.ipv4_option.option= 31;//1 byte
hdr.ipv4_option.swid = 1; //3 bits
hdr.ipv4_option.flow_packet_count = (bit<16>) meta.min_count;//2 bytes
hdr.ipv4_option.packets_in_queue = (bit<10>) var_deq_qdepth;
hdr.ipv4_option.hitter = meta.marker; //1 bit
hdr.ipv4_option.queue_timedelta = var_deq_timedelta;
hdr.ipv4_option.packet_length= (bit<18>)var_packet_length;
hdr.ipv4_option.optionLength =  12; //total number of byte
```

P4 Program at Switch at S1U Interface

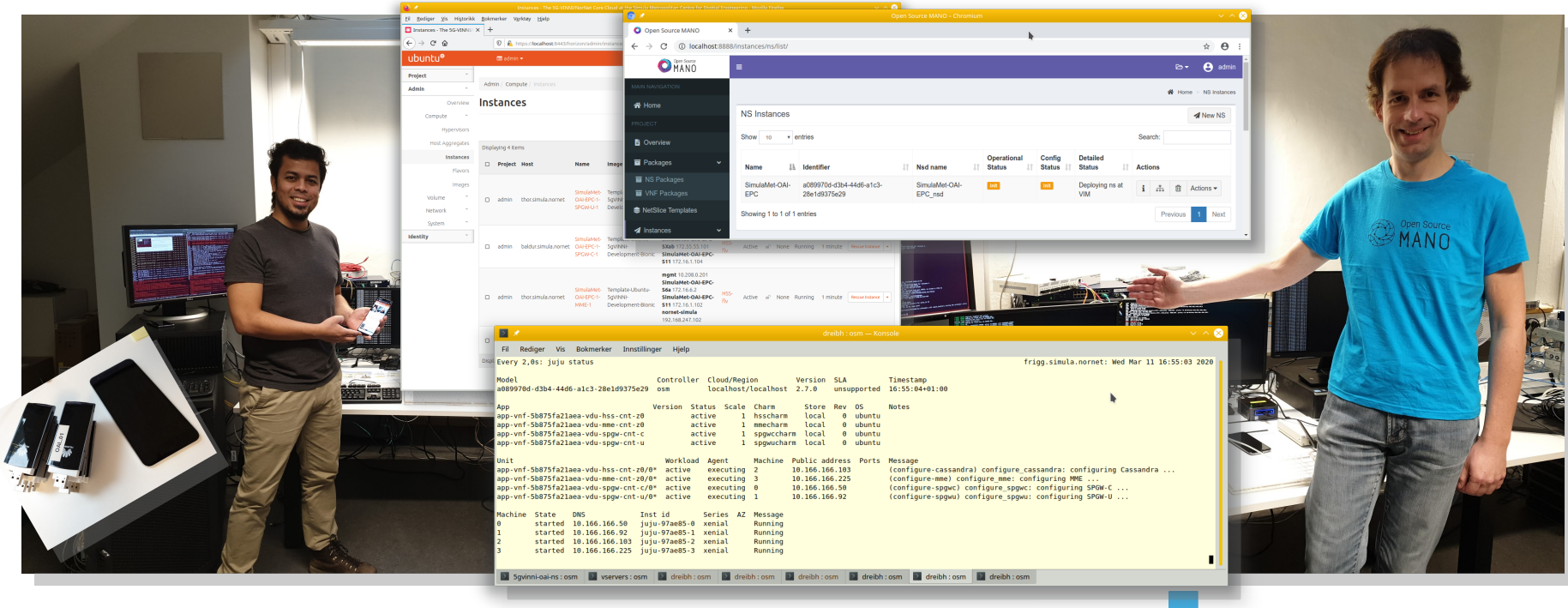


Collector collects the data for further analysis

```
###[ Ethernet ]###
  dst       = fa:16:3e:47:c4:89
  src       = fa:16:3e:a8:17:95
  type      = IPv4
###[ IP ]###
     version   = 4
     ihl       = 5
     tos       = 0x0
     len       = 1406
     id        = 19522
     flags     = DF
     frag      = 0
     ttl       = 30
     proto     = gre
     chksum    = 0xf489
     src       = 10.208.0.214
     dst       = 10.208.0.16
     \options   \
###[ GRE ]###
        chksum_present= 0
        routing_present= 0
        key_present= 0
        seqnum_present= 0
        strict_route_source= 0
        recursion_control= 0
        flags     = 0
        version   = 0
        proto     = IPv4
```

```
###[ IP ]###
           version   = 4
           ihl       = 8
           tos       = 0x0
           len       = 1382
           id        = 19522
           flags     = DF
           frag      = 0
           ttl       = 64
           proto     = udp
           chksum    = 0xcde1
           src       = 12.1.1.2
           dst       = 10.1.1.96
           \options   \
            |###[ TELEMETRY ]###
            |  copy_flag = 1
            |  optclass  = control
            |  option    = 31
            |  length    = 12
            |  swid      = 1
            |  flow_packet_count= 6762
            |  packets_in_queue= 0
            |  queue_timedelta= 49
            |  hitter    = 0
            |  packet_length= 1420
```

Live Demo!


Open Source MANO
NS Instances
Instances
dreibh : osm — Konsole

Any Questions?

Thomas Dreibholz

dreibh@simula.no, https://www.simula.no/people/dreibh

Andrés Felipe Ocampo

andres@simula.no, https://www.simula.no/people/andres

Mah-Rukh Fida

mahrukhfida@simula.no, https://www.simula.no/people/mahrukhfida

This project has received funding from the EU's Horizon 2020 research and innovation programme under grant agreement No 815279.