OSC RIC release J: KPM_REPORTING_PERIOD_AND_PER_FALLBACK

Date: 2025/12/25

Last modified: 2025/12/29

- OS: Ubuntu 20.04 desktop
- Kernel: 5.15.0-139-generic
- OSC RIC release J
  - https://docs.o-ran-sc.org/projects/o-ran-sc-ric-plt-ric-dep/en/latest/installation-guides.html#installing-near-realtime-ric-in-ric-cluster
  - Installation steps: Refer to WiSDON RIC_20251217.pptx
    - sudo apt install git
    - sudo apt install net-tools
    - sudo apt update -y && sudo apt upgrade –y
    - https://docs.o-ran-sc.org/projects/o-ran-sc-ric-plt-ric-dep/en/latest/installation-guides.html
    - git clone https://gerrit.o-ran-sc.org/r/ric-plt/ric-dep
    - (chatGPT helps, k8s installation error) sudo apt update
    - (chatGPT helps, k8s installation error) sudo apt install -y linux-headers-$(uname -r) conntrack
    - # install kubernetes, kubernetes-CNI, helm and docker
    - cd ric-dep/bin
    - sudo ./install_k8s_and_helm.sh
    -
    - # install chartmuseum into helm and add ric-common templates
    - sudo ./install_common_templates_to_helm.sh
    -
    - sudo groupadd docker
    - sudo usermod -aG docker $USER
    - # install the RIC
    - cd ric-dep/bin
    - sudo ./install -f ../RECIPE_EXAMPLE//example_recipe_oran_j_release.yaml
- E2sim: Release K
  - https://gerrit.o-ran-sc.org/r/gitweb?p=sim%2Fe2-interface.git;a=summary
- KPIMON: master
  - https://gerrit.o-ran-sc.org/r/c/ric-app/kpimon-go/+/12141
- RC: v1.0.4
  - https://gerrit.o-ran-sc.org/r/c/ric-app/rc/+/11051

For the time being, the architecture can be shown as the following illustrations

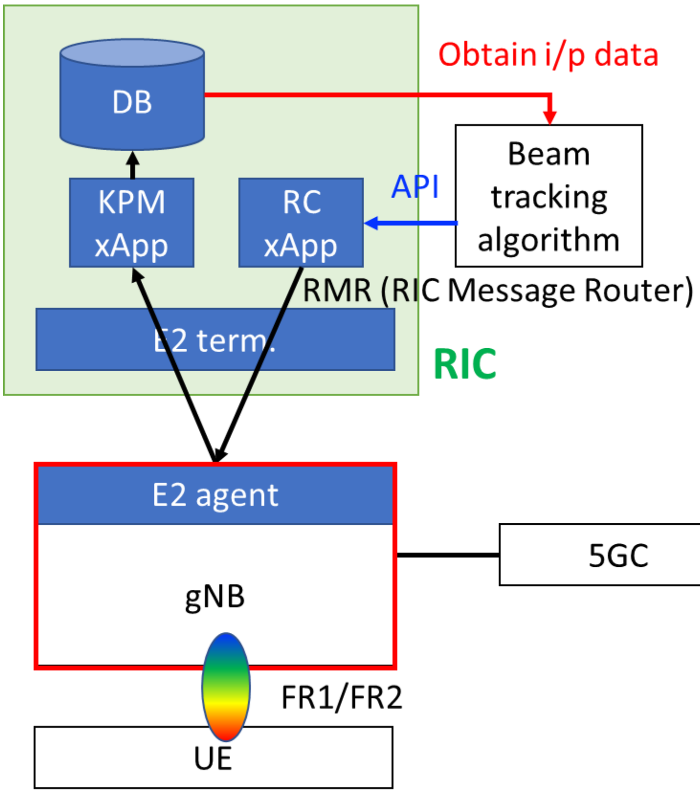
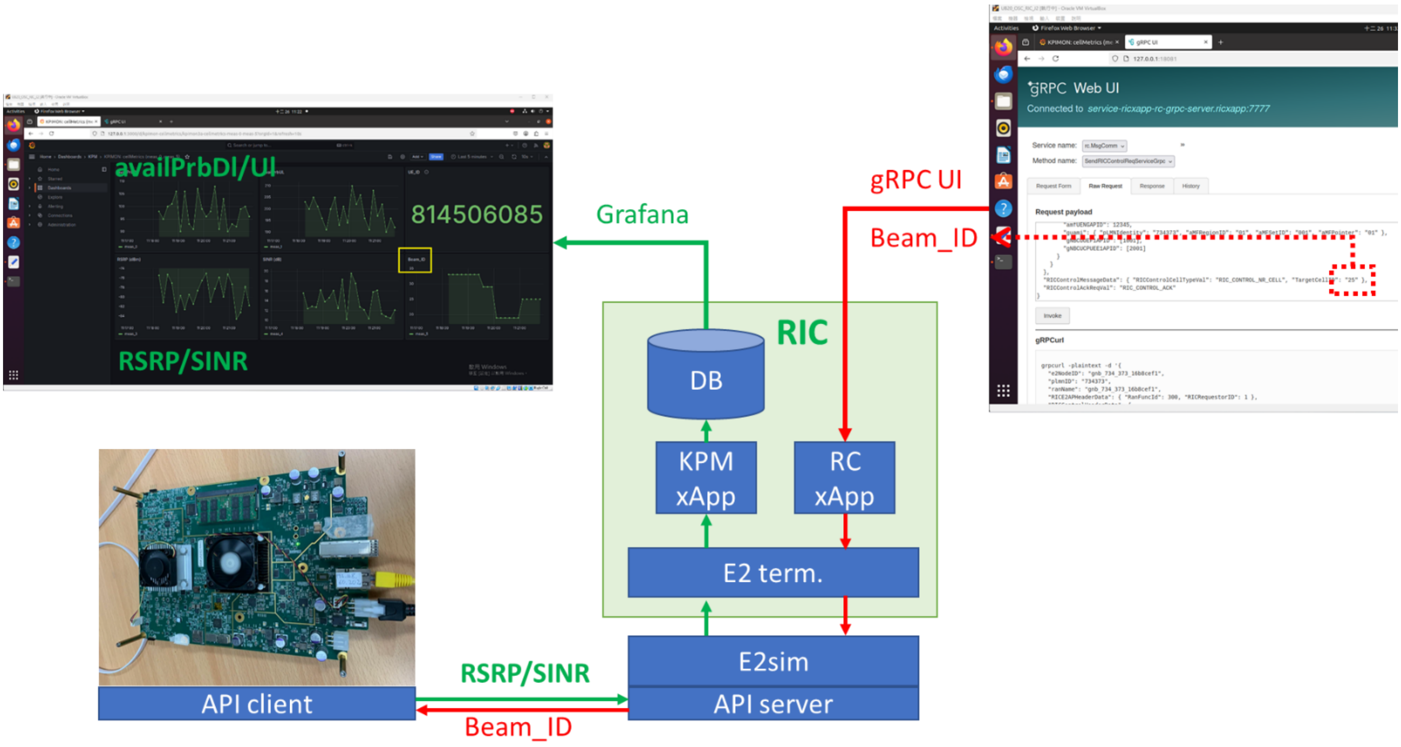


Illustration of integrating RIC, e2sim (e2agent), KPIMON and RC xApps

We make use of the gRPC API of RC xApp, using grpcui to simply demo pushing beam_ID (TargetCellID) to e2sim. The gNB (Synergy's board) obtains the beam_ID, randomly generates RSRP, SINR, and passes (via python 3.6) to API server on e2sim. Then, e2sim uploads the beam_ID, RSRP, SINR to e2term over e2 interface (SCTP, indications). We have the KPIMON xApp subscribe the e2sim, receive and decode the indications, and store in an influxDB. We install Grafana to show the data in influxDB. The whole procedure can be illustrated as the following illustration.



```
        "amfUENGAPID": 12345,
        "guami": { "pLMNIdentity": "734373", "aMFRegionID": "01", "aMFSetID": "001", "aMFPointer": "01" },
        "gNBCUUEF1APID": [1001],
        "gNBCUCPUEE1APID": [2001]
      }
    }
  },
  "RICControlMessageData": { "RICControlCellTypeVal": "RIC_CONTROL_NR_CELL", "TargetCellID": "25" },
  "RICControlAckReqVal": "RIC_CONTROL_ACK"
}
```

```
grpcurl -plaintext -d '{
  "e2NodeID": "gnb_734_373_16b8cef1",
  "plmnID": "734373",
  "ranName": "gnb_734_373_16b8cef1",
  "RICE2APHeaderData": { "RanFuncId": 300, "RICRequestorID": 1 },
```

Demo of RIC, e2sim, KPIMON, RC xApps and Synergy's board

Making use of gRPC API is just for now. Next we are going to deploy our own beam tracking xApp, making use of the RC E2AP/E2SM and ASN.1, which can be shown as the following illustration.

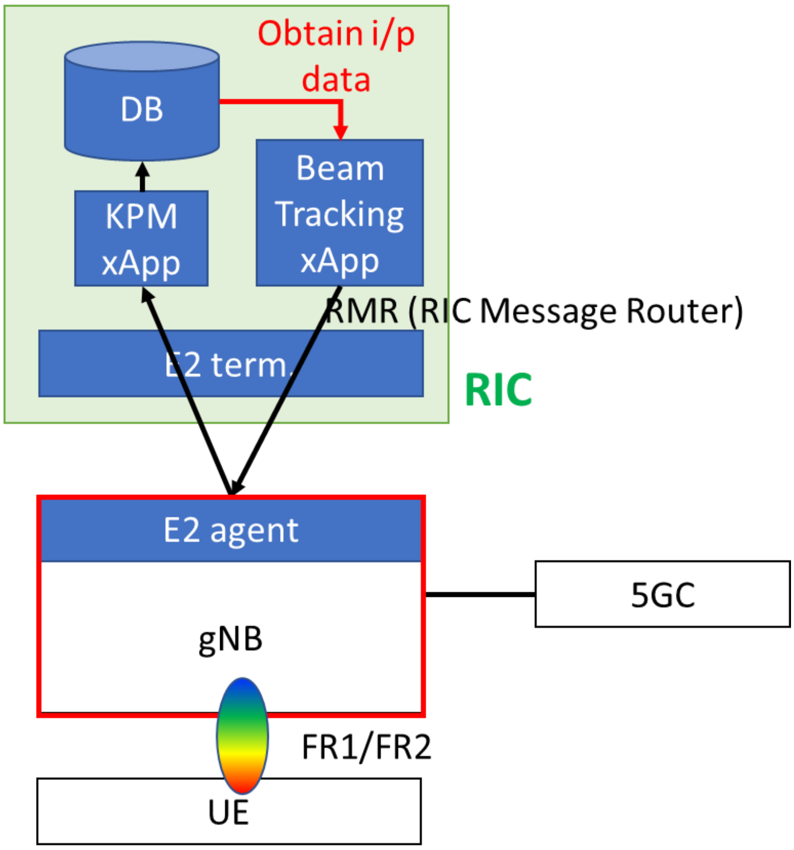


Illustration of beam tracking xApp, integrating with RIC, e2sim and KPIMON xApp

# KPM 訂閱週期 (reportingPeriod) 與 PER 編碼不一致 (APER/UPER) 的修正說明

本文件用「程式碼實際狀態」確認你描述的修正是否正確，並明確指出：

- `reportingPeriod` 的解析位置（Subscription Request 的 `EventTriggerDefinition`）

- e2sim 為什麼原本只送一次、現在如何做到週期性送 `RIC_INDICATION`

- APER / UPER 是什麼、以及本 repo 內「APER → UPER fallback」實作在哪裡、怎麼做

## 1) 你的印象是否正確？（結論）

是的，整體方向正確，而且目前 repo 的 `kpm_callbacks.cpp` 內確實存在：

- **解析 Subscription Request 內 `ricEventTriggerDefinition`**（E2SM-KPM EventTriggerDefinition Format1 的 `reportingPeriod`）

- **解碼時先用 APER（Aligned PER）嘗試，失敗再用 UPER（Unaligned PER）嘗試**（也就是你說的「APER → UPER fallback」）

- **`run_report_loop()` 外層迴圈**：讓 e2sim 依 `reportingPeriod` 週期性讀取 `reports.json` 並送出 `RIC_INDICATION`

## 2) `reportingPeriod` 定義在哪？e2sim 要怎麼拿到它？

### 2.1 Subscription Request 的來源

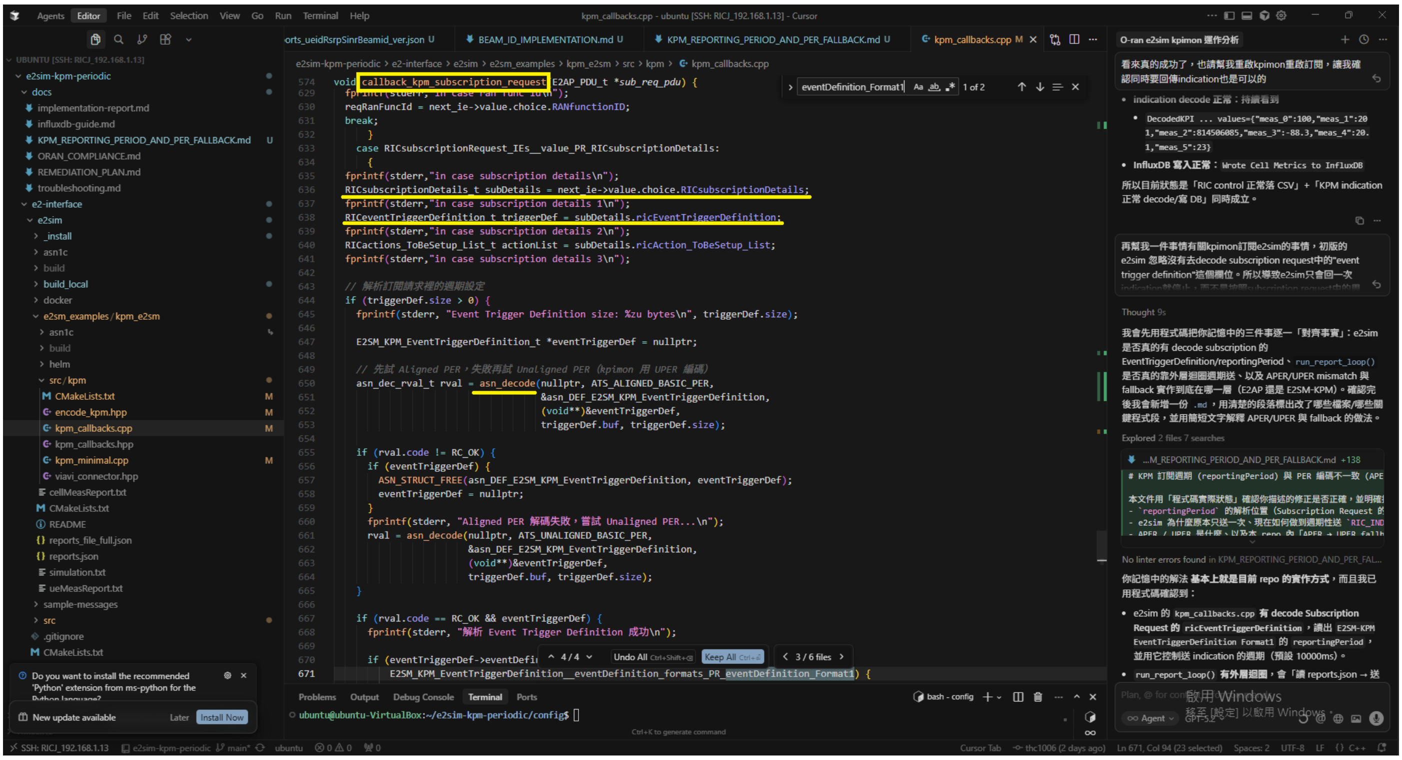
在 E2AP `RICsubscriptionRequest` 中，`RICsubscriptionDetails` 內含：

- `ricEventTriggerDefinition`（`RICeventTriggerDefinition_t`）：這個欄位在 KPM Service Model 裡承載 `E2SM-KPM-EventTriggerDefinition`

### 2.2 e2sim 解析位置（重點檔案/函數）

檔案：`e2-interface/e2sim/e2sm_examples/kpm_e2sm/src/kpm/kpm_callbacks.cpp`

函數：`callback_kpm_subscription_request(E2AP_PDU_t *sub_req_pdu)`



kpm_callbacks.cpp - ubuntu [SSH: RICJ_192.168.1.13] - Cursor
e2sim-kpm-periodic > e2-interface > e2sim > e2sm_examples > kpm_e2sm > src > kpm > kpm_callbacks.cpp
eventDefinition_Format1
1 of 2
O-ran e2sim kpimon 運作分析

```
void callback_kpm_subscription_request(E2AP_PDU_t *sub_req_pdu) {
  fprintf(stderr,"in case ran func id\n");
  reqRanFuncId = next_ie->value.choice.RANfunctionID;
  break;
    }
   case RICsubscriptionRequest_IEs__value_PR_RICsubscriptionDetails:
    {
  fprintf(stderr,"in case subscription details\n");
  RICsubscriptionDetails_t subDetails = next_ie->value.choice.RICsubscriptionDetails;
  fprintf(stderr,"in case subscription details 1\n");
  RICeventTriggerDefinition_t triggerDef = subDetails.ricEventTriggerDefinition;
  fprintf(stderr,"in case subscription details 2\n");
  RICactions_ToBeSetup_List_t actionList = subDetails.ricAction_ToBeSetup_List;
  fprintf(stderr,"in case subscription details 3\n");

  // 解析訂閱請求裡的週期設定
  if (triggerDef.size > 0) {
    fprintf(stderr, "Event Trigger Definition size: %zu bytes\n", triggerDef.size);

    E2SM_KPM_EventTriggerDefinition_t *eventTriggerDef = nullptr;

    // 先試 Aligned PER，失敗再試 Unaligned PER (kpimon 用 UPER 編碼)
    asn_dec_rval_t rval = asn_decode(nullptr, ATS_ALIGNED_BASIC_PER,
                                     &asn_DEF_E2SM_KPM_EventTriggerDefinition,
                                     (void**)&eventTriggerDef,
                                     triggerDef.buf, triggerDef.size);

    if (rval.code != RC_OK) {
      if (eventTriggerDef) {
        ASN_STRUCT_FREE(asn_DEF_E2SM_KPM_EventTriggerDefinition, eventTriggerDef);
        eventTriggerDef = nullptr;
      }
      fprintf(stderr, "Aligned PER 解碼失敗，嘗試 Unaligned PER...\n");
      rval = asn_decode(nullptr, ATS_UNALIGNED_BASIC_PER,
                        &asn_DEF_E2SM_KPM_EventTriggerDefinition,
                        (void**)&eventTriggerDef,
                        triggerDef.buf, triggerDef.size);
    }

    if (rval.code == RC_OK && eventTriggerDef) {
      fprintf(stderr, "解析 Event Trigger Definition 成功\n");

      if (eventTriggerDef->eventDefi
          E2SM_KPM_EventTriggerDefinition__eventDefinition_formats_PR_eventDefinition_Format1) {
```

```
ubuntu@ubuntu-VirtualBox:~/e2sim-kpm-periodic/config$
```

看來真的成功了，也請幫我重啟kpimon重啟訂閱，讓我確認同時要回傳indication也是可以的

- indication decode 正常：持續看到
  - `DecodedKPI ... values={"meas_0":100,"meas_1":201,"meas_2":814506085,"meas_3":-88.3,"meas_4":20.1,"meas_5":23}`
- InfluxDB 寫入正常：`Wrote Cell Metrics to InfluxDB`

所以目前狀態是「RIC control 正常落 CSV」+「KPM indication 正常 decode/寫 DB」同時成立。

再幫我一件事情有關kpimon訂閱e2sim的事情，初版的e2sim 忽略沒有去decode subscription request中的"event trigger definition"這個欄位。所以導致e2sim只會回一次

Thought 9s

我會先用程式碼把你記憶中的三件事逐一「對齊事實」：e2sim 是否真的有 decode subscription 的 EventTriggerDefinition/reportingPeriod、`run_report_loop()` 是否真的靠外層迴圈週期送、以及 APER/UPER mismatch 與 fallback 實作到底在哪一層（E2AP 還是 E2SM-KPM）。確認完後我會新增一份 `.md`，用清楚的段落標出改了哪些檔案/哪些關鍵程式段，並用簡短文字解釋 APER/UPER 與 fallback 的做法。

Explored 2 files 7 searches

...M_REPORTING_PERIOD_AND_PER_FALLBACK.md +138

```
# KPM 訂閱週期 (reportingPeriod) 與 PER 編碼不一致 (APE

本文件用「程式碼實際狀態」確認你描述的修正是否正確，並明確
- `reportingPeriod` 的解析位置 (Subscription Request 的
- e2sim 為什麼原本只送一次、現在如何做到週期性送 `RIC_IND
```

No linter errors found in KPM_REPORTING_PERIOD_AND_PER_FAL...

你記憶中的解法 **基本上就是目前 repo 的實作方式**，而且我已用程式碼確認到：

- e2sim 的 `kpm_callbacks.cpp` **有 decode Subscription Request 的** `ricEventTriggerDefinition`，讀出 `E2SM-KPM EventTriggerDefinition Format1` 的 `reportingPeriod`，並用它控制送 indication 的週期（預設 10000ms）。
- `run_report_loop()` **有外層迴圈**，會「讀 reports.json → 送

Agent
GPT-5.2
啟用 Windows
移至 [設定] 以啟用 Windows。
Ln 671, Col 94 (23 selected) Spaces: 2 UTF-8 LF C++

在 `RICsubscriptionRequest_IEs__value_PR_RICsubscriptionDetails` 的 case 內：

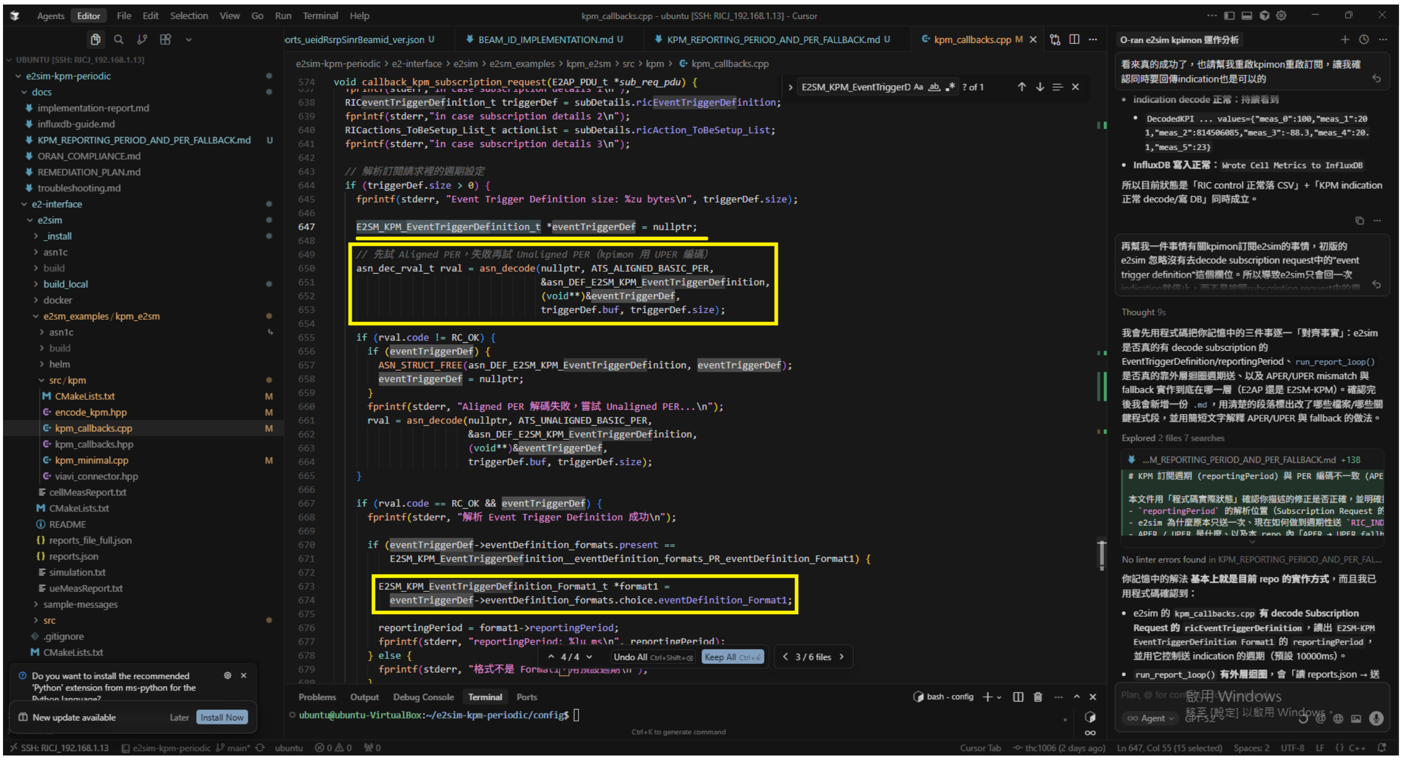
1. 取出 `RICsubscriptionDetails_t subDetails`

2. 取出 `RICeventTriggerDefinition_t triggerDef = subDetails.ricEventTriggerDefinition`

3. 用 ASN.1 將 `triggerDef.buf/size` decode 成 `E2SM_KPM_EventTriggerDefinition_t`

4. 若是 `eventDefinition_Format1`，就讀出 `format1->reportingPeriod`（單位 ms）

5. 若解碼失敗或格式不是 Format1，使用 default `reportingPeriod = 10000 ms`


kpm_callbacks.cpp - ubuntu [SSH: RICJ_192.168.1.13] - Cursor
orts_ueidRsrpSinrBeamid_ver.json U
BEAM_ID_IMPLEMENTATION.md U
KPM_REPORTING_PERIOD_AND_PER_FALLBACK.md U
kpm_callbacks.cpp M
e2sim-kpm-periodic > e2-interface > e2sim > e2sm_examples > kpm_e2sm > src > kpm > kpm_callbacks.cpp
E2SM_KPM_EventTriggerD
? of 1
4 / 4
Undo All
Keep All
3 / 6 files
Problems
Output
Debug Console
Terminal
Ports
bash - config
Ln 647, Col 55 (15 selected)
Spaces: 2
UTF-8
LF
C++

```
void callback_kpm_subscription_request(E2AP_PDU_t *sub_req_pdu) {
  RICeventTriggerDefinition_t triggerDef = subDetails.ricEventTriggerDefinition;
  fprintf(stderr,"in case subscription details 2\n");
  RICactions_ToBeSetup_List_t actionList = subDetails.ricAction_ToBeSetup_List;
  fprintf(stderr,"in case subscription details 3\n");

  // 解析訂閱請求裡的週期設定
  if (triggerDef.size > 0) {
    fprintf(stderr, "Event Trigger Definition size: %zu bytes\n", triggerDef.size);

    E2SM_KPM_EventTriggerDefinition_t *eventTriggerDef = nullptr;

    // 先試 Aligned PER，失敗再試 Unaligned PER (kpimon 用 UPER 編碼)
    asn_dec_rval_t rval = asn_decode(nullptr, ATS_ALIGNED_BASIC_PER,
                                     &asn_DEF_E2SM_KPM_EventTriggerDefinition,
                                     (void**)&eventTriggerDef,
                                     triggerDef.buf, triggerDef.size);

    if (rval.code != RC_OK) {
      if (eventTriggerDef) {
        ASN_STRUCT_FREE(asn_DEF_E2SM_KPM_EventTriggerDefinition, eventTriggerDef);
        eventTriggerDef = nullptr;
      }
      fprintf(stderr, "Aligned PER 解碼失敗，嘗試 Unaligned PER...\n");
      rval = asn_decode(nullptr, ATS_UNALIGNED_BASIC_PER,
                        &asn_DEF_E2SM_KPM_EventTriggerDefinition,
                        (void**)&eventTriggerDef,
                        triggerDef.buf, triggerDef.size);
    }

    if (rval.code == RC_OK && eventTriggerDef) {
      fprintf(stderr, "解析 Event Trigger Definition 成功\n");

      if (eventTriggerDef->eventDefinition_formats.present ==
          E2SM_KPM_EventTriggerDefinition__eventDefinition_formats_PR_eventDefinition_Format1) {

        E2SM_KPM_EventTriggerDefinition_Format1_t *format1 =
          eventTriggerDef->eventDefinition_formats.choice.eventDefinition_Format1;

        reportingPeriod = format1->reportingPeriod;
        fprintf(stderr, "reportingPeriod: %lu ms\n", reportingPeriod);
      } else {
        fprintf(stderr, "格式不是 Format1，用預設週期\n");
```

```
ubuntu@ubuntu-VirtualBox:~/e2sim-kpm-periodic/config$
```

O-ran e2sim kpimon 運作分析

看來真的成功了，也請幫我重啟kpimon重啟訂閱，讓我確認同時要回傳indication也是可以的

- indication decode 正常：持續看到
  - `DecodedKPI ... values={"meas_0":100,"meas_1":201,"meas_2":814506085,"meas_3":-88.3,"meas_4":20.1,"meas_5":23}`
- **InfluxDB 寫入正常**：`Wrote Cell Metrics to InfluxDB`

所以目前狀態是「RIC control 正常落 CSV」+「KPM indication 正常 decode/寫 DB」同時成立。

再幫我一件事情有關kpimon訂閱e2sim的事情，初版的e2sim 忽略沒有去decode subscription request中的"event trigger definition"這個欄位。所以導致e2sim只會回一次

Thought 9s

我會先用程式碼把你記憶中的三件事逐一「對齊事實」：e2sim 是否真的有 decode subscription 的 EventTriggerDefinition/reportingPeriod、`run_report_loop()` 是否真的靠外層迴圈週期送、以及 APER/UPER mismatch 與 fallback 實作到底在哪一層（E2AP 還是 E2SM-KPM）。確認完後我會新增一份 `.md`，用清楚的段落標出改了哪些檔案/哪些關鍵程式段，並用簡短文字解釋 APER/UPER 與 fallback 的做法。

Explored 2 files 7 searches

...M_REPORTING_PERIOD_AND_PER_FALLBACK.md +138

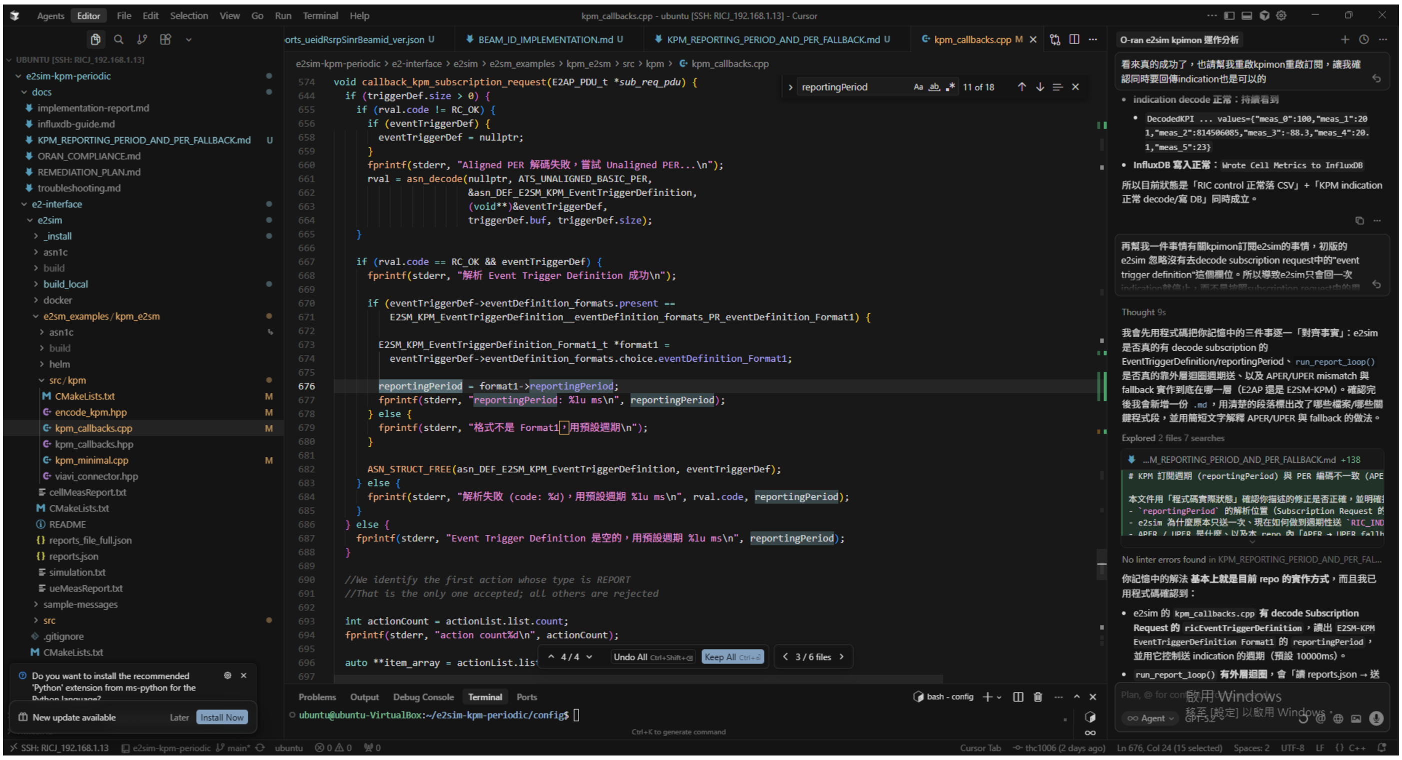
```
# KPM 訂閱週期 (reportingPeriod) 與 PER 編碼不一致 (APE

本文件用「程式碼實際狀態」確認你描述的修正是否正確，並明確
- `reportingPeriod` 的解析位置 (Subscription Request 的
- e2sim 為什麼原本只送一次、現在如何做到週期性送 `RIC_IND
```

No linter errors found in KPM_REPORTING_PERIOD_AND_PER_FAL...

你記憶中的解法 **基本上就是目前 repo 的實作方式**，而且我已用程式碼確認到：

- e2sim 的 `kpm_callbacks.cpp` **有 decode Subscription Request 的** `ricEventTriggerDefinition`，讀出 `E2SM-KPM EventTriggerDefinition Format1` 的 `reportingPeriod`，並用它控制送 indication 的週期（預設 10000ms）。
- `run_report_loop()` **有外層迴圈**，會「讀 reports.json → 送


kpm_callbacks.cpp - ubuntu [SSH: RICJ_192.168.1.13] - Cursor
reportingPeriod
11 of 18
4 / 4
Undo All
Keep All
3 / 6 files
Ln 676, Col 24 (15 selected)

```
void callback_kpm_subscription_request(E2AP_PDU_t *sub_req_pdu) {
  if (triggerDef.size > 0) {
    if (rval.code != RC_OK) {
      if (eventTriggerDef) {
        eventTriggerDef = nullptr;
      }
      fprintf(stderr, "Aligned PER 解碼失敗，嘗試 Unaligned PER...\n");
      rval = asn_decode(nullptr, ATS_UNALIGNED_BASIC_PER,
                        &asn_DEF_E2SM_KPM_EventTriggerDefinition,
                        (void**)&eventTriggerDef,
                        triggerDef.buf, triggerDef.size);
    }

    if (rval.code == RC_OK && eventTriggerDef) {
      fprintf(stderr, "解析 Event Trigger Definition 成功\n");

      if (eventTriggerDef->eventDefinition_formats.present ==
          E2SM_KPM_EventTriggerDefinition__eventDefinition_formats_PR_eventDefinition_Format1) {

        E2SM_KPM_EventTriggerDefinition_Format1_t *format1 =
          eventTriggerDef->eventDefinition_formats.choice.eventDefinition_Format1;

        reportingPeriod = format1->reportingPeriod;
        fprintf(stderr, "reportingPeriod: %lu ms\n", reportingPeriod);
      } else {
        fprintf(stderr, "格式不是 Format1，用預設週期\n");
      }

      ASN_STRUCT_FREE(asn_DEF_E2SM_KPM_EventTriggerDefinition, eventTriggerDef);
    } else {
      fprintf(stderr, "解析失敗 (code: %d)，用預設週期 %lu ms\n", rval.code, reportingPeriod);
    }
  } else {
    fprintf(stderr, "Event Trigger Definition 是空的，用預設週期 %lu ms\n", reportingPeriod);
  }

  //We identify the first action whose type is REPORT
  //That is the only one accepted; all others are rejected

  int actionCount = actionList.list.count;
  fprintf(stderr, "action count%d\n", actionCount);

  auto **item_array = actionList.list
```

```
ubuntu@ubuntu-VirtualBox:~/e2sim-kpm-periodic/config$
```

O-ran e2sim kpimon 運作分析

看來真的成功了，也請幫我重啟kpimon重啟訂閱，讓我確認同時要回傳indication也是可以的

- indication decode 正常：持續看到
  - `DecodedKPI ... values={"meas_0":100,"meas_1":201,"meas_2":814506085,"meas_3":-88.3,"meas_4":20.1,"meas_5":23}`
- **InfluxDB 寫入正常**：`Wrote Cell Metrics to InfluxDB`

所以目前狀態是「RIC control 正常落 CSV」+「KPM indication 正常 decode/寫 DB」同時成立。

再幫我一件事情有關kpimon訂閱e2sim的事情，初版的e2sim 忽略沒有去decode subscription request中的"event trigger definition"這個欄位。所以導致e2sim只會回一次

Thought 9s

我會先用程式碼把你記憶中的三件事逐一「對齊事實」：e2sim 是否真的有 decode subscription 的 EventTriggerDefinition/reportingPeriod、`run_report_loop()` 是否真的靠外層迴圈週期送、以及 APER/UPER mismatch 與 fallback 實作到底在哪一層（E2AP 還是 E2SM-KPM）。確認完後我會新增一份 `.md`，用清楚的段落標出改了哪些檔案/哪些關鍵程式段，並用簡短文字解釋 APER/UPER 與 fallback 的做法。

Explored 2 files 7 searches

...M_REPORTING_PERIOD_AND_PER_FALLBACK.md +138

No linter errors found in KPM_REPORTING_PERIOD_AND_PER_FAL...

你記憶中的解法 **基本上就是目前 repo 的實作方式**，而且我已用程式碼確認到：

- e2sim 的 `kpm_callbacks.cpp` **有 decode Subscription Request 的** `ricEventTriggerDefinition`，讀出 `E2SM-KPM EventTriggerDefinition Format1` 的 `reportingPeriod`，並用它控制送 indication 的週期（預設 10000ms）。
- `run_report_loop()` **有外層迴圈**，會「讀 reports.json → 送

## 3) 為什麼「初版只回一次 indication 就停止」？現在怎麼做到週期性？

### 3.1 目前 repo 的週期送法（重點）

檔案：`e2-interface/e2sim/e2sm_examples/kpm_e2sm/src/kpm/kpm_callbacks.cpp`

函數：`run_report_loop(...)`

現在的行為是：

- `run_report_loop()`有外層迴圈（在stop_flag為true的情況下持續執行）

- 每一輪會打開`reports.json`（或fallback到viavi connector）

- 逐行parse JSON（JSON-lines格式）

- encode並送出`RIC_INDICATION`

- **讀完一輪後sleep `reportingPeriodMs`**，再讀下一輪

也就是：**「讀檔 → 送indication → sleep(週期) → 重複」**

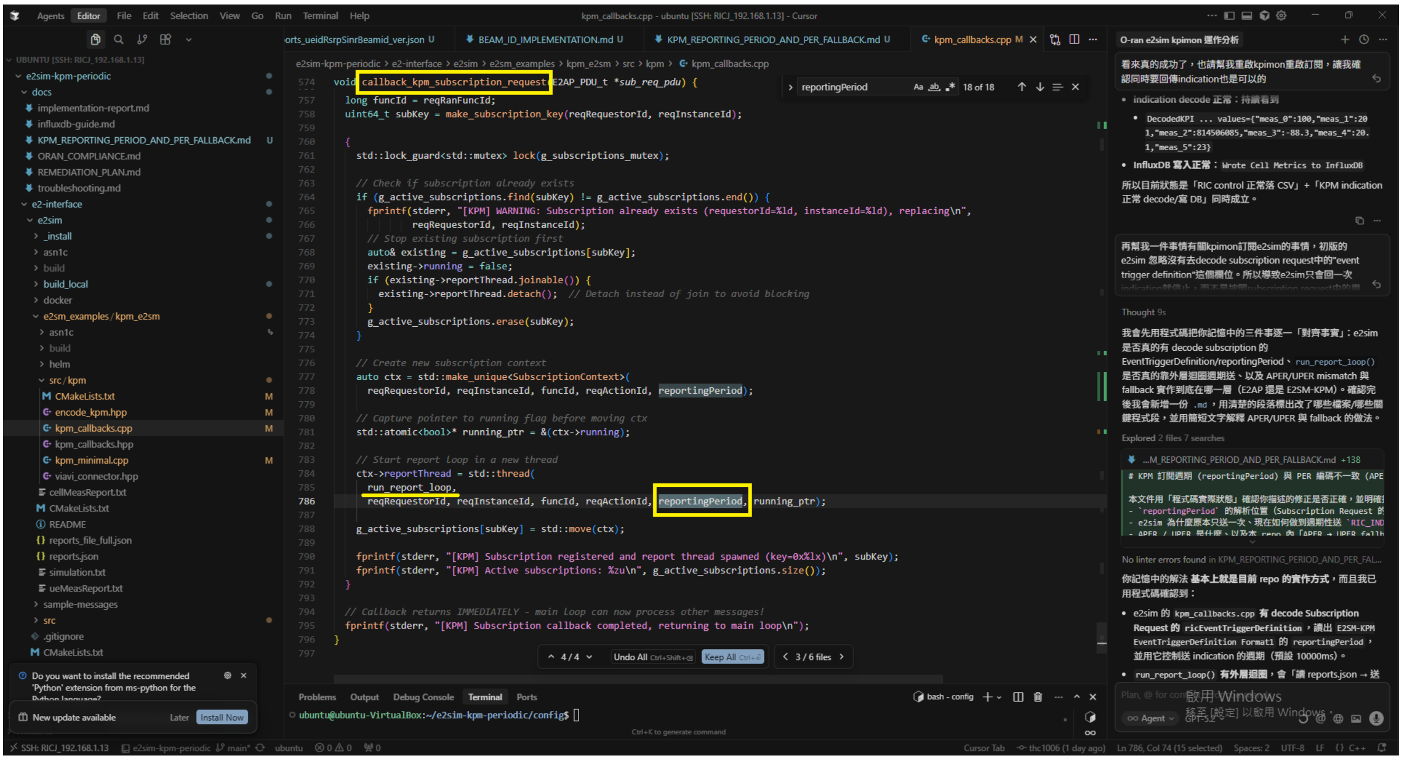
### 3.2 為什麼callback不應該阻塞？

在`callback_kpm_subscription_request(...)`中：

- 會先回`RICsubscriptionResponse`

- 再把`run_report_loop(...)`放到thread裡跑（避免阻塞main loop繼續處理後續E2AP訊息）

這點對穩定性很關鍵：不然e2sim的SCTP receive loop可能被subscription callback卡住。


kpm_callbacks.cpp - ubuntu [SSH: RICJ_192.168.1.13] - Cursor
e2sim-kpm-periodic > e2-interface > e2sim > e2sm_examples > kpm_e2sm > src > kpm > kpm_callbacks.cpp
reportingPeriod
18 of 18
4 / 4
Undo All
Keep All
3 / 6 files
Problems
Output
Debug Console
Terminal
Ports
bash - config
Ctrl+K to generate command
O-ran e2sim kpimon 運作分析
Agent
GPT-5.2
Ln 786, Col 74 (15 selected)
Spaces: 2
UTF-8
LF
C++

```
void callback_kpm_subscription_request(E2AP_PDU_t *sub_req_pdu) {
  long funcId = reqRanFuncId;
  uint64_t subKey = make_subscription_key(reqRequestorId, reqInstanceId);

  {
    std::lock_guard<std::mutex> lock(g_subscriptions_mutex);

    // Check if subscription already exists
    if (g_active_subscriptions.find(subKey) != g_active_subscriptions.end()) {
      fprintf(stderr, "[KPM] WARNING: Subscription already exists (requestorId=%ld, instanceId=%ld), replacing\n",
              reqRequestorId, reqInstanceId);
      // Stop existing subscription first
      auto& existing = g_active_subscriptions[subKey];
      existing->running = false;
      if (existing->reportThread.joinable()) {
        existing->reportThread.detach();  // Detach instead of join to avoid blocking
      }
      g_active_subscriptions.erase(subKey);
    }

    // Create new subscription context
    auto ctx = std::make_unique<SubscriptionContext>(
      reqRequestorId, reqInstanceId, funcId, reqActionId, reportingPeriod);

    // Capture pointer to running flag before moving ctx
    std::atomic<bool>* running_ptr = &(ctx->running);

    // Start report loop in a new thread
    ctx->reportThread = std::thread(
      run_report_loop,
      reqRequestorId, reqInstanceId, funcId, reqActionId, reportingPeriod, running_ptr);

    g_active_subscriptions[subKey] = std::move(ctx);

    fprintf(stderr, "[KPM] Subscription registered and report thread spawned (key=0x%lx)\n", subKey);
    fprintf(stderr, "[KPM] Active subscriptions: %zu\n", g_active_subscriptions.size());
  }

  // Callback returns IMMEDIATELY - main loop can now process other messages!
  fprintf(stderr, "[KPM] Subscription callback completed, returning to main loop\n");
}
```

```
ubuntu@ubuntu-VirtualBox:~/e2sim-kpm-periodic/config$
```

看來真的成功了，也請幫我重啟kpimon重啟訂閱，讓我確認同時要回傳indication也是可以的

- indication decode 正常：持續看到
  - `DecodedKPI ... values={"meas_0":100,"meas_1":201,"meas_2":814506085,"meas_3":-88.3,"meas_4":20.1,"meas_5":23}`
- **InfluxDB 寫入正常**：`Wrote Cell Metrics to InfluxDB`

所以目前狀態是「RIC control 正常落 CSV」+「KPM indication 正常 decode/寫 DB」同時成立。

再幫我一件事情有關kpimon訂閱e2sim的事情，初版的e2sim 忽略沒有去decode subscription request中的"event trigger definition"這個欄位。所以導致e2sim只會回一次

Thought 9s

我會先用程式碼把你記憶中的三件事逐一「對齊事實」：e2sim 是否真的有 decode subscription 的 EventTriggerDefinition/reportingPeriod、`run_report_loop()` 是否真的靠外層迴圈週期送、以及 APER/UPER mismatch 與 fallback 實作到底在哪一層（E2AP 還是 E2SM-KPM）。確認完後我會新增一份 `.md`，用清楚的段落標出改了哪些檔案/哪些關鍵程式段，並用簡短文字解釋 APER/UPER 與 fallback 的做法。

Explored 2 files 7 searches

...M_REPORTING_PERIOD_AND_PER_FALLBACK.md +138

```
# KPM 訂閱週期 (reportingPeriod) 與 PER 編碼不一致 (APE

本文件用「程式碼實際狀態」確認你描述的修正是否正確，並明確
- `reportingPeriod` 的解析位置 (Subscription Request 的
- e2sim 為什麼原本只送一次、現在如何做到週期性送 `RIC_IND
```

No linter errors found in KPM_REPORTING_PERIOD_AND_PER_FAL...

你記憶中的解法 **基本上就是目前 repo 的實作方式**，而且我已用程式碼確認到：

- e2sim 的 `kpm_callbacks.cpp` **有 decode Subscription Request 的** `ricEventTriggerDefinition`，讀出 `E2SM-KPM EventTriggerDefinition Format1` 的 `reportingPeriod`，並用它控制送 indication 的週期（預設 10000ms）。
- `run_report_loop()` **有外層迴圈**，會「讀 reports.json → 送

## 4) APER / UPER 是什麼？為什麼會「編碼不一致」？

### 4.1 名詞解釋（簡短版）

- **PER**：ASN.1 的 Packed Encoding Rules（X.691）

- **APER (Aligned PER)**：Aligned Packed Encoding Rules（對齊版 PER）

- **UPER (Unaligned PER)**：Unaligned Packed Encoding Rules（不對齊版 PER）

兩者都是 PER，但位元對齊規則不同，因此：

- **同一個 ASN.1 物件**若用 APER vs UPER 編碼，產生的 byte stream 不同

- 若一端用 UPER encode、另一端用 APER decode（或反過來），就可能 decode 失敗

### 4.2 本 repo 的實際情況（你描述的「kpimon 用 UPER，e2sim 用 APER」）

在 kpimon 端（C wrapper）：

- `E2SM_KPM_EventTriggerDefinition` 的 encode 使用 `uper_encode_to_buffer(...)`

  - 表示它送出的 `EventTriggerDefinition` 是 **UPER**

在 e2sim 端（KPM callback）:

- 對 `EventTriggerDefinition` 先用 `ATS_ALIGNED_BASIC_PER`（Aligned PER）嘗試 decode

- 若失敗，再用 `ATS_UNALIGNED_BASIC_PER`（Unaligned PER / UPER）decode

## 5) 「APER → UPER fallback」機制是怎麼加入的？

### 5.1 實作位置（e2sim 端）

檔案：`e2-interface/e2sim/e2sm_examples/kpm_e2sm/src/kpm/kpm_callbacks.cpp`

位置：`callback_kpm_subscription_request()` 解析 `ricEventTriggerDefinition` 時

核心邏輯（文字描述）:

1. `asn_decode(... ATS_ALIGNED_BASIC_PER ...)`

2. 如果 `rval.code != RC_OK`：

    - free 已分配的 `eventTriggerDef`

    - log：「Aligned PER 解碼失敗，嘗試 Unaligned PER...」

    - `asn_decode(... ATS_UNALIGNED_BASIC_PER ...)`

3. 如果 decode 成功，取 `reportingPeriod`

4. 否則用 default 10000ms

這就是你說的「APER → UPER fallback」(先試 APER/Aligned，再試 UPER/Unaligned)。

### 5.2 類似的fallback（kpimon端）

kpimon的C wrapper也有類似策略：在decode `IndicationHeader/IndicationMessage`時先試`ATS_ALIGNED_BASIC_PER`，失敗再試`ATS_UNALIGNED_BASIC_PER`。

## 6) 如何驗證（建議觀察點）

### 6.1 驗證`reportingPeriod`有被解析

看e2sim log是否出現類似：
- `Event Trigger Definition size: ...`
- `解析Event Trigger Definition成功`
- `reportingPeriod: <N> ms`

若decode失敗會看到：

- `Aligned PER解碼失敗，嘗試Unaligned PER...`

- 或最後fallback：

  - `解析失敗 ... 用預設週期10000 ms`

### 6.2 驗證週期性送indication

看e2sim log是否反覆出現：

- `讀完reports.json，等<reportingPeriod> ms後繼續...`

以及kpimon是否持續出現：

- `DecodedKPI: ... interval=5s values=...`
- `Wrote Cell Metrics to InfluxDB`