

# 新一代 Telemetry 遙測網管系統設計與實作 -以 TWAREN 骨幹網路為例

陳俊傑 林孟璋

財團法人國家實驗研究院 國家高速網路與計算中心

{jjchen, kent}@niar.org.tw

## 摘要

傳統以 SNMP 技術收集資料的網管系統有著資料傳遞容易延遲及大型骨幹網路網管資料維護不易問題，本文將說明新一代 Telemetry 遙測網管技術設計優點，如何解決傳統 SNMP 網管系統的問題，以 TWAREN 骨幹網路為例，實作 Telemetry 遙測網管系統，內容包含資料結構、系統架構設計、資料收集與分析方式、網管監控與告警方式並利用 Grafana 軟體視覺化網管監控資料，比較 Telemetry 遙測網管資料與傳統 SNMP 網管資料，說明兩者的差異性與優缺點。

關鍵詞：Telemetry，網路管理，SNMP，TWAREN

## 1. 前言

隨著雲端服務與多樣態的網際網路應用興起，提供底層網路服務的骨幹網路規模越來越大，管理這些網路設備與網路服務的網路管理系統架構與資料亦越發龐大，在 RFC 6632[1]與思科(Cisco)的網路管理白皮書[2]中提到網路管理系統的五大管理元素，Fault、Configuration、Accounting、Performance and Security management(FCAPS)，其中網管資料的收集大部分仰賴 Simple Network Management Protocol(SNMP)協定[3]，SNMP 已經存在了三十年，其運行方式沒有改變以滿足現代網路的監控需求。真正的問題在於 SNMP 的執行速度，SNMP 帶來的問題癥結在於其基本操作行為的一部分，因此 SNMP 提供甚少/沒有改進的空間，Telemetry 遙測技術執行速可解決 SNMP 即時監控與效能上的問題。

### ● 執行速度和即時監控需求

SNMP 使用 PULL Model - GetBulk / GetNext

操作，這些操作通過從一次又一次定期的請求詢問設備。此外，如果大型表不能容納在一個資料包中，則需要多個請求。這是導致 SNMP 速度減慢的最大瓶頸，而且傳送的資料通常以分鐘為單位超過某個時間因素。這種延遲無法滿足現代較大骨幹網路監控要求；Telmetry 遙測技術使用使用 PUSH 模型，並且固有不受上面列出的限制，因為它知道以什麼時間間隔向誰傳送什麼資料。它只需一次查詢即可收集資料，並使用預構建的內部模板實現超高速的內部操作，因而能夠以極少的時間交付更多資料。

### ● 頭外的開銷和缺少最佳化選項

由 SNMP 提取的資料實際上是以內部資料結構的形式儲存的，需要由設備節點進行內部轉換。這

是網路節點將內部資料結構對映為 SNMP 格式的幕後額外工作；Telemetry 直接提取內部資料結構，並且在將資料傳送出去之前執行最小處理，從而以儘可能最少的時間和精力提供最更新的資料。

### ● 工作負載的線性性質

我們在同一時間輪詢相同的準確資料，每個額外的輪詢站都會導致設備節點上的負載增加。從多個輪詢設並行訪問同一 MIB 可能導致響應速度較慢和 CPU 利用率較高；如果多個目的地需要相同的資料，則 Telemetry 遙測需要一次提取資料並複製資料包，PUSH 推式模式在速度和規模方面勝過 SNMP PULL 模式。

SNMP 與 Telemetry 資料收集的方法及運作方式比較如表1[4]。

表1 SNMP 與 Telemetry 資料收集與運作方式比較

| SNMP      | Telemetry |
|-----------|-----------|
| 非即時資訊     | 即時資訊      |
| 可擴充性差     | 高度可擴展     |
| PULL Mode | PUSH Mode |
| 非自動化      | 自動化-Ready |

Telemetry 遙測網管技術具備監控即時性、系統效能較佳的優點及適用於大型網路架構，本文將於第2章節說明 Telemetry 遙測資料內容與傳輸架構，Telemetry 遙測網管系統架構設計與分析，第3章節描述在 TWAREN 骨幹網路實作 Telemetry 遙測網管系統的設定與結果，第4章節分析及比較 TWAREN 骨幹網路的 SNMP 與 Telemetry 的網管資料，第5章節說明結論與未來系統展望。

## 2. Telemetry 網管系統設計與分析

根據 RFC 9232 Network Telemetry Framework [5]定義，Telemetry 框架分三個模組，第一個模組是關於資料顯示，資料顯示是資訊參考與組織方式，第二個模組是關於編碼，每個取樣間隔，遙測都會將以上測量資料轉換為可線上中進行序列化的格式，第三個是關於運輸的這是用於在裝置之間傳輸資料的協定棧，模組功能與其主要元件說明如表2。

表2 Telemetry 框架模組功能

| 模組功能    | 元件                        |
|---------|---------------------------|
| 1. 資料表示 | YANG 資料模型                 |
| 2. 編碼   | GPB/GPB "self-describing" |
| 3. 傳輸   | TCP/gRPC                  |

YANG 資料模型在 RFC 790[6]被定義為「資料建模語言，用於為網路管理協定對配置資料、狀態資料和通知進行建模。」由於 YANG 與典型程式語言體系結構的分離性，它可以與多種工具進行互動。YANG 建模資料結構是圍繞模組和子模組的概念構建的，它以樹狀的方式定義資料的層次。YANG 模型有多種來源可供使用，其中以下三種來源被認為是主要來源：原生型號/思科特定、OpenConfig、IETF。本文設計與實作採用思科特定 YANG 資料模型，可從 GitHub[7]中尋找思科特定型號設備支援的 YANG 資料模型，內容範例如圖1。

```

submodule Cisco-IOS-XR-pfi-im-cmd-oper-sub1 {
    belongs-to Cisco-IOS-XR-pfi-im-cmd-oper {
        prefix Cisco-IOS-XR-pfi-im-cmd-oper;
    }
    import ietf-inet-types {
        prefix inet;
    }
    import ietf-yang-types {
        prefix yang;
    }
    import Cisco-IOS-XR-types {
        prefix xr;
    }
    include Cisco-IOS-XR-pfi-im-cmd-oper-sub2 {
        revision-date 2017-06-26;
    }
    organization
        "Cisco Systems, Inc.";
    contact
        "Cisco Systems, Inc.
        Customer Service
        Postal: 170 West Tasman Drive
        San Jose, CA 95134
        Tel: +1 800 553-NETS
        E-mail: cs-yang@cisco.com";
    description
        "This submodule contains a collection of YANG definition
        for Cisco IOS-XR pfi-im-cmd package operational data.
        Copyright (c) 2013-2017 by Cisco Systems, Inc.
        All rights reserved.";
    revision 2017-06-26 {
        description
            "Change identifiers to be more readable.";
    }
    revision 2017-05-01 {
        description
            "Fixing backward compatibility error in module.";
    }
    revision 2016-12-18 {
        description
            "Description updated.";
    }
    revision 2015-11-09 {
        description
            "IOS XR 6.0 revision.";
    }
    typedef Stats-counter {
        type enumeration {
            enum "stats-counter-rate" {
                value 0;
                description
                    "stats counter rate";
            }
            enum "stats-counter-uint32" {
                value 1;
                description
                    "stats counter uint32";
            }
            enum "stats-counter-uint64" {
                value 2;
                description
                    "stats counter uint64";
            }
            enum "stats-counter-generic" {
                value 3;
                description
                    "stats counter generic";
            }
        }
    }
}

```

圖1 思科 YANG Model 範例

編碼將資料轉換為可通過網路傳輸的格式。當接收器對資料進行解碼時，它擁有原始資料的語義相同的副本。GPB(Google Protocol Buffers)最終被思科採用，因為它提高了編碼操作的效率和速度。GPB 有兩種形式作為遙測流的編碼選項：GPB-“compact”、GPB-“self-describing”。兩種 GPB 遙測格式之間的主要區別在於它們如何表示與編碼遙測資料流中的金鑰。TWAREN 系統

實作採用 GPB-“self-describing”方式。

傳輸方式，Telmetry 遙測提供三種傳輸協定提供選項，TCP、gRPC、UDP。預設情況下，TCP 是遙測預定義的傳輸方法，因為它是可靠的，而且非常容易配置為選項。gRPC 是一個現代開源框架，設計為可在任何環境中運行。它構建在 HTTP/2之上，並提供一組增強和豐富的功能。在 Telmetry 遙測架構設計中，還需要資料解碼、資料收集與監控模組如設計架構圖2。

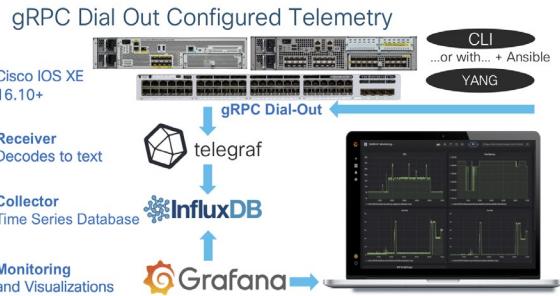


圖2 Telemetry 系統設計架構圖

使用 Telegraf 來解碼網路設備送出來的 Telemetry 串流資料，Telegraf 是一套由 InfluxData 開發的開放跨平台資料收集代理(agent)，主要用於把各種系統、應用程式、資料庫、網路裝置等的度量指標(metrics)和事件(events)收集起來，然後送往時間序列資料庫(Time-Series Database，TSDB)或其他儲存/處理端點；使用 InfluxDB 來儲存 Telemetry 網管資料，InfluxDB 是一款專門用來儲存、查詢與分析時間序列資料的開放資料庫，其特性為可高效寫入與讀取資料，適合網管資料這類型的時間序列串流資料；使用 Grafana 來進行資料視覺化與網管監控，Grafana 是一套開放的資料視覺化與監控平台，其優點為方便快速建立儀表板(Dashboard)、支援多種資料來源、支援即時監控(秒級)更新的即時圖，告警與觸發通知。

### 3. TWAREN Telemetry 網管系統實作

#### 3.1 網路設備 Telmetry 設定

TWAREN 骨幹網路設備我 ASR9K 系列，作業系統為 IOS-XR，設定 Telemetry 有幾個步驟如下，

##### 3.1.1 建立 destination-group，

要建立 destination-group，您需要知道 collector 收集器的 IPv4 地址和為此應用程式提供服務的通訊埠並指定網路設備和收集器上相對應的協定和編碼，範例如下，

```

telemetry model-driven
destination-group DG1
vrf mgmt.--> 指定用於與收集器網路地址通訊的虛擬路由和轉發(VRF)
address-family ipv4 192.168.3.75 port 57800
encoding self-describing-gpb

```

```

protocol tcp
encoding self-describing-gpb
編碼選項:
RP/0/RP0/CPU0:ASR9006(config-model-
driven-dest-addr)#encoding ?
  gpb      GPB encoding
  json     JSON encoding
  self-describing-gpb Self describing GPB
  encoding
協定選項:
RP/0/RP0/CPU0:ASR9006(config-model-driven-
dest-addr)#protocol ?
  grpc  gRPC
  tcp   TCP
  udp   UDP
3.1.2 建立 sensor-group
列出我們感想要監控的 sensor path，xpath 是
思科設備中用來定位 YANG 模型中某個節點
的語法，它告訴 Telemetry Engine 要將哪一類
的網管資料送出。類似傳統 SNMP MIB 檔案
中的 OID 有其對應的監控標的，我們可以從
思科提供的 Yansuite 系統[8]來找到 OID 相對
應的 xpath。設定範例如下，
sensor-group Mem-Used
  sensor-path Cisco-IOS-XR-nto-misc-shmem-
    oper:memory-summary/nodes/node/summary
3.1.3 建立 subscribe
向設訂閱傳輸哪種網管感應串流資料至哪個
目的地 collector，設定範例如下，
subscription CPU-Util
  sensor-group-id Monitor-CPU sample-interval
    60000(串流資料輸出頻率，單位 ms)
  destination-id DG2
3.1.4 驗證設定與通訊是否正確
顯示訂閱狀態、資料傳輸、collector 收取狀
況，設定範例與顯示如下，
show telemetry model-driven subscription CPU-
Util
Subscription: CPU-Util
-----
State: ACTIVE
Sensor groups: Id: Monitor-CPU
Sample Interval: 60000 ms
Sensor Path:Cisco-IOS-XR-wdysmon-fd-
oper:system-monitoring/cpu-utilization
Sensor Path State: Resolved
Destination Groups:
  Group Id: DG2
  Destination IP: 192.168.3.75
  Destination Port: 57800
  Destination Vrf: mgmt(0x60000002)
  Encoding: self-describing-gp
  Transport: tcp
  State: Active
  No TLS
  Total bytes sent: 182467506731
  Total packets sent: 1395073
  Last Sent time: 2025-09-07

```

此次實作監控標的 OID 值與 XPATH 對應如表3。

表3 OID 與 XPATH 對應表

| OID名稱            | OID編號                           | OID描述                             | XPATH   |
|------------------|---------------------------------|-----------------------------------|---|
| cpmCPUTotal      | 1.3.6.1.4.1.9.9.1<br>09.1.1.1.1 | CPU忙碌百分比                          | Cisco-IOS-XR- wdysmon-fd- oper:system- monitoring/cpu- utilization                                |
| cempMempoolUse d | 1.3.6.1.4.1.9.9.2<br>21.1.1.1.7 | 指示物理實體上的 應用程式當前正在 使用的記憶體池中 的位元組數。 | Cisco-IOS-XR- nto- misc-shmem- oper:memory- summary/nodes/node/ summary                           |
| ifOutOctets      | 1.3.6.1.2.1.2.2.1.<br>16        | 從介面傳輸出去的 八位元總數。                   | Cisco-IOS-XR-infra- statsd-oper:infra- statistics/interfaces/in- terface/latest/generic- counters |

## 3.2 Telegraf、InfluxDB 與 Grafana 設定

### 3.2.1 Telegraf

要從 Cisco ASR9K 路由器接收介面的 Byte Count 收集串流遙測數據，本系統使用 Telegraf 配置的 MDT 輸入插件(Model-driven Telemetry Input Plugin)，設備主動送出 Telemetry 資料，傳送至接收端定好的固定埠 (Port,57800)。

#### Step 1 (Telegraf configuration file)

Telegraf 設定檔(MDT)，處理設備傳送過來的
串流資料

```

# MDT input Configuration
[[inputs.cisco_telemetry_mdt]]
## Telemetry transport can be "tcp" or "grpc"
transport = "tcp"
## Address and port to host telemetry listener
service_address = "0.0.0.0:57800"
## Grpc Maximum Message Size, default is
4MB, increase the size. This is
stored as a uint32, and limited to 4294967295.
max_msg_size = 4000000
Step 2 寫入 InfluxDB 的設定檔
# Output Plugin Configuration (Example for In-
fluxDB)
[[outputs.influxdb]]
urls = ["http://192.168.3.75:8085"]
database = "telemetryasr"
username = "<your influxdb account>"
password = "<your influxdb password>"
```

### 3.2.2 InfluxDB

InfluxDB 是一個時間序列(Time Series Data-
base, TSDB)資料庫，專門處理 Metrics (度量
數據)、Logs、IoT 資料流的儲存引擎。其
InfluxQL 類似於關連式資料庫(RDBMS)的
SQL 語法。

例如：

```

SELECT mean("usage_cpu") FROM "cpu"
WHERE time > now() - 1h GROUP BY
time(1m)

```

這語法以 `usage_cpu` 做為存取對象，且可以透過制訂的函數予以計算，例如上述函數 `mean`，即使用平均來計算取出的值。位於 `FROM` 之後，藉以表示存取資料表哪一個資料表，範例中命名為 `cpu`，在 InfluxDB 中以量測表(Measurement)名詞來稱呼。最後於 `where` 語法中，給予條件予以過濾。例如，我們在 Grafana 中以存取，存取介面流量時，就會寫下如下的查詢語法

```

SELECT NON_NEGATIVE_DERIVA-
TIVE(mean("bytes_received"), 1m) * 8 /
1000000 as "Rx Mbps" FROM "Cisco-IOS-XR-
infra-statsd-oper:infra-statistics/interfaces/interface/latest/generic-counters" WHERE "inter-
face_name" = 'TenGigE0/1/0' and
"source"='TWAREN-TN-ASR9006-01' AND
time > now() - 2880m GROUP BY time(60s)

```

另外，我們亦有優化資料庫查詢：

- `NON_NEGATIVE_DERIVATIVE`, (非負導數)函數，藉以防止在圖表中出現負值的情況，因為有時設備 Byte Count 會出現重設(Reset)成0的狀況，在計算流量時會算出現異常數值，藉此更準確地呈現實際介面流量使用率。
- 使用多重的查詢條件，將想要查詢某台設備上的某一個介面 Byte Count 予以正確取得。語法可以在 `WHERE` 之後以關鍵字 `AND` 依序並連達成。Ex, 查詢 TWAREN-TN-ASR9006-01 且 TenGigE0/1/0 這個介面。SELECT mean("usage\_cpu") FROM "cpu" WHERE time > now() - 1h GROUP BY time(1m) check-interval = "30"

### 3.2.3 Grafana 設定

新增資料來源 Data Source: InfluxDB，如圖3。

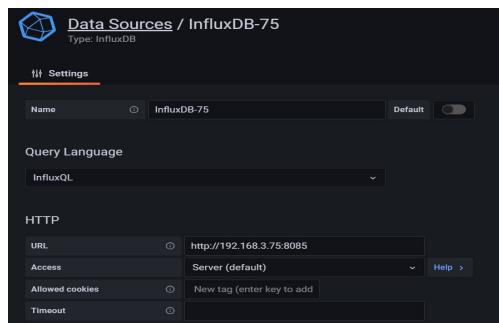


圖3 Grafana Data Source 設定

## 3.3 資料傳輸驗證

### 3.3.1 網路設備端驗證，指令範例如圖4。

show telemetry model-driven summary，顯示 Destination group、Sensor Group、Subscription 狀態正常 Active。

```

RP/0/RSP0/CPU0/TWAREN-TN-ASR9006-01#show telemetry model-driven summary
Sun Sep 7 21:32:41.736 cst
Subscriptions   Total: 6      Active: 5      Paused: 0
Destination Groups Total: 4
Destinations    gpc-tls: 0      gpc-nontls: 0      tcp: 4      udp: 0
                  dialin: 0      Active: 2      Sessions: 10      Connecting: 6
Sensor Groups   Total: 6
Sensor Paths    Total: 5      Active: 6      Not Resolved: 0
Max Sensor Paths: 1000
Max Containers per path: 16

```

圖4 網路設備顯示 Telemetry 設定狀態

### 3.3.2 InfluxDB 資料庫資料驗證

顯示資料庫內有建立系統愈收集的監控標的 Sensor Group XPATH 相對應的資料表如圖5。

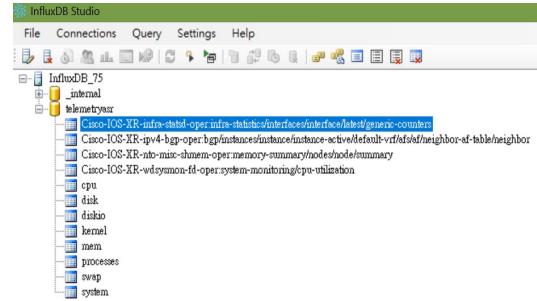


圖5 InfluxDB XPATH 對應資料表

Query 資料庫，確認 Telemetry 遙測串流資料有進資料庫如圖6。

| #  | time                | cpu_usage | available_fqg | bandwidth_peak | bandwidth_peak_pct | bytes_received | bytes_sent   | current_bandwidth | cr_recon | framing_error | idle_packet_pct | rx_packets |
|----|---------------------|-----------|---------------|----------------|--------------------|----------------|--------------|-------------------|----------|---------------|-----------------|------------|
| 1  | 2025/07/25 07:25:00 | 0         | 0             | 0              | 0                  | 0              | 0            | 0                 | 0        | 0             | 0               | 1          |
| 2  | 2025/07/25 07:25:00 | 0         | 0             | 0              | 0                  | 0              | 0            | 0                 | 0        | 0             | 0               | 1          |
| 3  | 2025/07/25 07:25:00 | 0         | 0             | 0              | 0                  | 0              | 0            | 0                 | 0        | 0             | 0               | 1          |
| 4  | 2025/07/25 07:25:00 | 0         | 0             | 0              | 0                  | 0              | 0            | 0                 | 0        | 0             | 0               | 1          |
| 5  | 2025/07/25 07:25:00 | 0         | 0             | 0              | 0                  | 0              | 0            | 0                 | 0        | 0             | 0               | 1          |
| 6  | 2025/07/25 07:25:00 | 0         | 0             | 0              | 0                  | 0              | 0            | 0                 | 0        | 0             | 0               | 1          |
| 7  | 2025/07/25 07:25:00 | 0         | 0             | 0              | 0                  | 0              | 0            | 0                 | 0        | 0             | 0               | 1          |
| 8  | 2025/07/25 07:25:00 | 0         | 0             | 0              | 0                  | 0              | 0            | 0                 | 0        | 0             | 0               | 1          |
| 9  | 2025/07/25 07:25:00 | 0         | 0             | 0              | 0                  | 0              | 0            | 0                 | 0        | 0             | 0               | 1          |
| 10 | 2025/07/25 07:25:00 | 0         | 0             | 0              | 0                  | 0              | 0            | 0                 | 0        | 0             | 0               | 1          |
| 11 | 2025/07/25 07:25:00 | 0         | 0             | 0              | 0                  | 0              | 0            | 0                 | 0        | 0             | 0               | 1          |
| 12 | 2025/07/25 07:25:00 | 0         | 0             | 11380          | 2676               | 402951213948   | 216681699695 | 5                 | 0        | 0             | 0               | 1          |
| 13 | 2025/07/25 07:25:00 | 0         | 0             | 6888           | 738                | 12063885558    | 1109514115   | 0                 | 0        | 0             | 0               | 1          |
| 14 | 2025/07/25 07:25:00 | 0         | 0             | 0              | 0                  | 0              | 0            | 0                 | 0        | 0             | 0               | 1          |

圖6 InfluxDB 資料 query 結果

## 4. Telemetry 與 SNMP 資料分析

TWAREN NOC 現階段使用的網管系統為 TWAREN 網管開發團隊針對 TWAREN 骨幹網路自行開發的100G整合式網管監控平台[9]，其底層資料收集係利用 SNMP 協定，由網管主機定期至骨幹網路設備收集監控標的並存放於 MariaDB 資料庫進行監控，為便於比較與分析 Telemetry 資料與 SNMP 資料的差異性，利用 Grafana 來同時呈現兩者相同監控標的資料與圖形如圖7，監控區間為2025/9/7 00:00-23:59 24小時，左邊為 Telemetry 1分鐘 PUSH 推送一次資料，右邊為 SNMP 5分鐘 PULL 查詢一次。

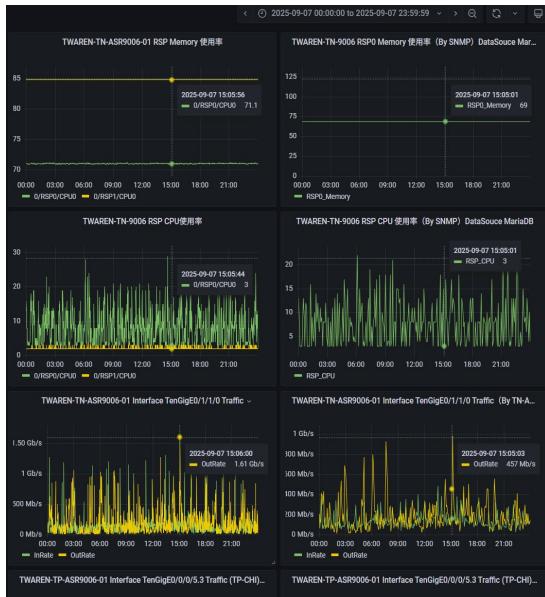


圖 7 Telemetry 與 SNMP 監控圖形(24小時)

Telemetry 產生 1438 筆資料，SNMP 產生 288 筆資料。Memory 記憶體使用率統計資料比較如表 4，可以發現因為 SNMP 的數據為五分鐘平均，而 Telemetry 為一分鐘平均，數據標準差較大，在網管監控上較容易監測出 Mirco-burst 障礙，這也是 Telemetry 的優點之一。

表 4 Memory Utilization 統計比較表

| Memory Utilization | No.  | MAX  | MIN | MEAN  | STD   |
|--------------------|------|------|-----|-------|-------|
| Telemetry          | 1438 | 71.2 | 71  | 71.07 | 0.056 |
| SNMP               | 288  | 69   | 69  | 69    | 0     |

Traffic Utilization 流量統計比較表如表 5，可以發現 Telemetry 資料 STD 標準差比較大，最大流量值亦比較大，在於 SNMP 為五分鐘平均，使的資料較為平均，無法較即時監測出流量的突波。

表 5 Traffic Utilization 統計比較表

| Traffic Utilization | No.  | MAX  | MIN  | MEAN   | STD     |
|---------------------|------|------|------|--------|---------|
| Telemetry           | 1438 | 1609 | 8.93 | 190.7  | 209.032 |
| SNMP                | 288  | 988  | 17.3 | 191.48 | 152.594 |

## 5. 結論與未來展望

Telemetry 遙測串流網管技術由設備主動推送串流資料，推送週期可由管理人員自訂，提升了系統傳輸效能與降低了傳統 SNMP Polling-based 對設備的負載，在資料處理上，提升資料顆粒度，可更即時發現異常，便於偵測 Microburst 微爆流，短時間的突波異常，在現行骨幹網路良流量越來越大狀況下，可更早發現網路攻擊與流量異常，未來甚至可導入人工智慧機器學習預測模型，例如，LSTM、RNN 模型來預測未來監控標的或流量，作為告警的基準。

## 參考文獻

- [1] RFC 6632，<https://datatracker.ietf.org/doc/html/rfc6632>
- [2] Network Management System: Best Practices White Paper，<https://www.cisco.com/c/en/us/support/docs/availability/high-availability/15114-NMS-bestpractice.html>
- [3] RFC 1157，<https://datatracker.ietf.org/doc/html/rfc1157>
- [4] ASR9K Model Driven Telemetry Whitepaper，<https://www.cisco.com/c/en/us/support/docs/routers/asr-9000-series-aggregation-services-routers/215321-asr9k-model-driven-telemetry-whitepaper.html>
- [5] RFC 9232，<https://datatracker.ietf.org/doc/html/rfc9232/>
- [6] RFC 7950，<https://datatracker.ietf.org/doc/html/rfc7950>
- [7] Cisco YANG Model，<https://github.com/YangModels/yang/tree/main/vendor/cisco>
- [8] Cisco YANG Suite，<https://developer.cisco.com/yangsuite/>
- [9] 鄭欣恬，"TWAREN 100G 整合式網管監控平台之規劃與建置"，TANET2016 臺灣網際網路研討會，花蓮，2016