A YANG Data Model for External Event-driven Network Traffic Volume Prediction

Hongseok Jeon
Electronics and Telecommunications Research Institute
Daejeon, Korea
jeonhs@etri.re.kr

Seunghyun Yoon
Electronics and Telecommunications Research Institute
Daejeon, Korea
shpyoon@etri.re.kr

***Abstract*—Network traffic volume prediction systems require data inputs that extend beyond conventional network performance metrics. This paper presents a YANG data model for capturing and managing network-external events (NEEs)—events sourced from systems external to the network infrastructure that significantly influence traffic patterns. The proposed model defines a structured schema for representing NEEs, enabling their seamless integration as input data into traffic prediction and analysis frameworks, thereby enhancing the contextual basis for forecasting.**

***Index Terms*—Network traffic volume prediction, network-external events, YANG, data schema, streaming telemetry**

I. Introduction

Traditional network traffic volume prediction (NTVP) approaches primarily rely on network-internal performance metrics such as link utilization, flow counts, and latency measurements [1], [2]. While effective for short-term forecasting, these methods fail to capture the broader context provided by *network-external events* (NEEs), defined as events sourced from systems external to the network infrastructure (e.g., user-scheduled data backups, large-scale software updates, environmental sensor readings, and social media–driven activities) [3]. Recent studies have demonstrated that incorporating contextual information, including public event schedules and online activity trends, can enhance prediction accuracy and adaptability [4]–[6]. However, such approaches often employ ad hoc data handling, vendor-specific formats, and non-standardized structures, which hinder scalability and interoperability.

To address these limitations, this paper proposes a YANG-based data schema for representing diverse NEEs in a structured and extensible manner. The schema is designed to support established network management protocols—such as NETCONF, RESTCONF, and gNMI—thereby promoting cross-vendor interoperability. The model enables the consistent collection, classification, and management of NEE data for use as input to network traffic prediction or analysis frameworks. Fig. 1 illustrates the architecture of an NEE–driven NTVP system for transport networks. In this architecture, the proposed YANG schema defines the structure for NEE representation, while an information server aggregates events from heterogeneous sources. A T-SDN controller retrieves structured NEE data over gNMI using the Capabilities, Get, Set, and Subscribe RPCs. In particular, the Subscribe RPC supports streaming telemetry for real-time updates, delivering NEEs as atomic, ordered updates.

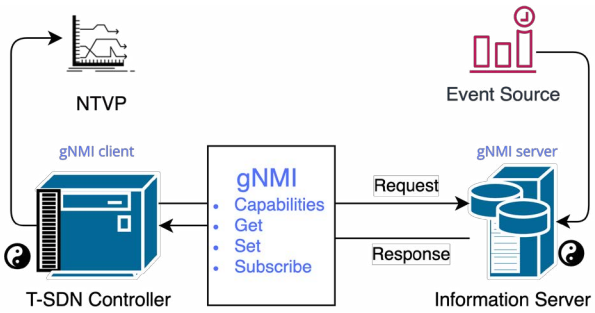

Fig. 1. Architecture of an NEE-driven NTVP system

This work's primary contribution is a YANG-based schema that formalizes the representation of NEEs in a structured and extensible manner.

II. Model Design

A. Reference Model and Design Rationale

The proposed YANG model draws architectural inspiration from the `openconfig-messages.yang` model [7], which standardizes message handling for system-generated log messages. While `openconfig-messages.yang` focuses on internal syslog events, our model targets NEEs that influence network traffic patterns. Both models adopt similar design patterns—configuration containers for filtering, state containers for operational data, and atomic telemetry extensions—however, the proposed model extends these concepts to accommodate heterogeneous data sourced from systems external to the network infrastructure.

B. Model Structure

The proposed model adopts a hierarchical structure grounded in three core YANG constructs: `identity` definitions, `typedef` specifications, and `grouping` abstractions.

As shown in Listing 1, the model first defines event source types and event criticality levels. The *event-message-source-type* `identity` serves as the foundation for extensible source

This work was supported by the Institute of Information and Communications Technology Planning and Evaluation (IITP) and funded by the Korea government (MSIT) under Grant No. RS-2021-II210851, On-demand data based network intelligence framework technology development.

classification, enabling vendors and operators to introduce custom source types without modifying the core module. This design facilitates integration of emerging data sources—such as IoT sensors and AI-driven analytics—into traffic prediction and analysis workflows.

The *event-message-criticality* `typedef` defines a four-level severity scale—ALERT, WARNING, NOTICE, and INFORMATIONAL—inspired by RFC 3164; we adopt a condensed four-level operational scale for simplicity. Each level is assigned an explicit numeric value (0–3) to ensure consistent priority ordering across implementations, thereby enabling automated filtering and prioritization of event messages based on operational significance.

```
identity event-message-source-type {
  description
    "Base identity for event message sources.
     Identities within this base identity are
     to be augmented by vendors.";
}
typedef event-message-criticality {
  type enumeration {
    enum ALERT {
      value 0;
      description
        "Alert: action must be taken immediately (0)";
    }
    enum WARNING {
      value 1;
      description
        "Warning: warning conditions (1)";
    }
    enum NOTICE {
      value 2;
      description
        "Notice: normal but significant condition (2)";
    }
    enum INFORMATIONAL {
      value 3;
      description
        "Informational: informational data (3)";
    }
  }
}
```

Listing 1. Definitions of event source type and criticality.

The overall schema is organized into three primary `groupings`, as illustrated in Listing 2. The first, *event-message-state*, defines operational data structures for NEEs. Within this grouping, the *event-message* `container` specifies five attributes: (i) *msg*, a flexible string payload that conveys NEE information in natural language form, enabling rich contextual descriptions such as scheduled service changes, sensor observations, and social media–derived insights; (ii) *priority*, the severity level as defined by *event-message-criticality*; (iii) *source-name*, a human-readable identifier for the originating system or service (e.g., "SeismicSensor-01", "TwitterTrendAPI"); (iv) *source-type*, an `identityref` to the base *event-message-source-type* `identity` that classifies the event source; and (v) *timestamp*, the creation time of the event message. This grouping applies the OpenConfig `telemetry-atomic` extension, ensuring that all elements of a data update are transmitted as an indivisible unit, which in practice avoids partial states and clarifies per-source update ordering.

The second `grouping`, *event-message-config*, defines configuration parameters, most notably the *criticality* threshold specifying the minimum severity level required for event transmission.

The third `grouping`, *network-external-data*, serves as the top-level `container` that integrates both configuration and state. The *config* `container` enables dynamic adjustment of the *criticality* threshold, whereas the *state* `container` provides read-only access to current settings and the latest event message. This organization preserves a strict separation between configuration and operational data, in accordance with OpenConfig design principles.

```
grouping event-message-state:
  +--ro event-message
     +--ro msg?            string
     +--ro priority?       event-message-criticality
     +--ro source-name?    string
     +--ro source-type?    identityref
     +--ro timestamp?      yang:date-and-time
grouping event-message-config:
  +-- criticality?   event-message-criticality
grouping network-external-data:
  +-- config
  |  +-- criticality?   event-message-criticality
  +--ro state
     +--ro criticality?     event-message-criticality
     +--ro event-message
        +--ro msg?            string
        +--ro priority?       event-message-criticality
        +--ro source-name?    string
        +--ro source-type?    identityref
        +--ro timestamp?      yang:date-and-time
```

Listing 2. Schema tree of groupings.

III. Conclusion

This paper introduced a YANG-based data model for representing NEEs that may affect network traffic patterns. By formalizing the structure of NEEs as a YANG-based schema, the model provides a consistent and interoperable input source for context-aware traffic prediction and analysis systems. Future work will focus on three directions: (i) developing machine learning algorithms optimized for the model's data structures; (ii) conducting performance evaluations to quantify its impact on prediction accuracy and interoperability; and (iii) exploring mechanisms for real-time correlation between NEEs and network performance.

References

[1] J. Lv, Y. Lin, M. Hou, Y. Li, Y. Gao, and W. Dong, "Accurate Bandwidth and Delay Prediction for 5G Cellular Networks," *ACM Transactions on Internet Technology*, vol. 25, no. 2, pp. 1–25, 2025.

[2] T. Otoshi, Y. Ohsita, M. Murata, Y. Takahashi, K. Ishibashi, and K. Shiomoto, "Traffic prediction for dynamic traffic engineering," *Computer Networks*, vol. 85, pp. 36–50, 2015.

[3] W. Jiang, "Cellular traffic prediction with machine learning: A survey," *Expert Systems with Applications*, vol. 201, p. 117163, 2022.

[4] X. Wang, Z. Zhou, F. Xiao, K. Xing, Z. Yang, Y. Liu, and C. Peng, "Spatio-Temporal Analysis and Prediction of Cellular Traffic in Metropolis," *IEEE Transactions on Mobile Computing*, vol. 18, no. 9, pp. 2190–2202, 2017.

[5] T. Kuber, I. Seskar, and N. Mandayam, "Traffic prediction by augmenting cellular data with non-cellular attributes," in *2021 IEEE wireless communications and networking conference (WCNC)*. IEEE, 2021, pp. 1–6.

[6] J. L. Bejarano-Luque, M. Toril, M. Fernández-Navarro, C. Gijón, and S. Luna-Ramírez, "A Deep-Learning Model for Estimating the Impact of Social Events on Traffic Demand on a Cell Basis," *IEEE Access*, vol. 9, pp. 71 673–71 686, 2021.

[7] (2024) openconfig-messages. OpenConfig. [Online]. Available: https://openconfig.net/projects/models/schemadocs/yangdoc/openconfig-messages.html