

Network Working Group
Internet-Draft
Intended status: Informational
Expires: 3 September 2026

D. King
Lancaster University
L. M. Contreras
Telefonica
B. Sipos
JHU/APL
L. Zhang
Huawei
2 March 2026

Time-Variant Routing (TVR) Requirements
draft-ietf-tvr-requirements-08

Abstract

Time-Variant Routing (TVR) refers to calculating a path or subpath through a network where the time of message transmission (or receipt) is part of the overall route computation. This means that, all things being equal, a TVR computation might produce different results depending on the time that the computation is performed without other detectable changes to the network topology or other cost functions associated with the route.

This document introduces requirements for the design and implementation of systems which perform TVR computations. It also explains different aspects of a TVR system which need to be considered during its design.

About This Document

This note is to be removed before publishing as an RFC.

Status information for this document may be found at
<https://datatracker.ietf.org/doc/draft-ietf-tvr-requirements/>.

Discussion of this document takes place on the Time Variant Routing Working Group mailing list (<mailto:tvr@ietf.org>), which is archived at <https://mailarchive.ietf.org/arch/browse/tvr/>. Subscribe at <https://www.ietf.org/mailman/listinfo/tvr/>.

Source for this draft and an issue tracker can be found at
<https://github.com/danielkinguk/tvr-requirements>.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <https://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 3 September 2026.

Copyright Notice

Copyright (c) 2026 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

Table of Contents

| | |
|--|----|
| 1. Introduction | 4 |
| 1.1. Conventions and Definitions | 4 |
| 2. Overview of Time-Variant Networks | 6 |
| 2.1. Resource Scheduling | 7 |
| 2.1.1. Schedule Domains | 7 |
| 2.1.2. Schedule Visibility | 8 |
| 2.1.3. Generation Locality | 9 |
| 2.1.4. Execution Locality | 9 |
| 2.1.5. Configuration and Operational State | 12 |
| 2.2. General Temporality | 12 |
| 2.2.1. Scope of Time-Variability | 12 |
| 2.2.2. Time Horizon | 13 |
| 2.2.3. Time Precision and Accuracy | 13 |
| 2.2.4. Time Synchronization and Margin | 14 |
| 2.2.5. Validity in a Schedule | 14 |
| 2.2.6. Periodicity in a Schedule | 15 |
| 2.2.7. Continuity in a Schedule | 15 |
| 2.2.8. Time-Overlap and Priority | 15 |
| 2.2.9. Property Value Interpolation | 16 |
| 2.2.10. Changes to Model State | 16 |

| | |
|---|----|
| 2.3. Topologies | 17 |
| 2.3.1. Nodes | 17 |
| 2.3.2. Termination Points | 17 |
| 2.3.3. Links | 18 |
| 2.3.4. Network Layering | 18 |
| 2.4. Routing Strategies | 18 |
| 2.4.1. Centralized | 19 |
| 2.4.2. Distributed | 19 |
| 2.4.3. Hybrid | 20 |
| 2.4.4. Constraints | 20 |
| 2.5. Integrity Considerations | 21 |
| 3. Time-Variant Use Case Requirements | 21 |
| 3.1. Resource Preservation Use Case | 21 |
| 3.2. Operating Efficiency Use Case | 22 |
| 3.3. Dynamic Reachability Use Case | 22 |
| 4. Requirements Summary | 23 |
| 4.1. Support the Identification and Advertisement of Entity Property Changes | 23 |
| 4.2. Support Proxy Advertisement | 24 |
| 4.3. Support Identification and Classification of Node Properties | 24 |
| 4.4. Support System Schedule and Time Interval Changes | 24 |
| 4.5. Support Appropriate Time Accuracy | 24 |
| 4.6. Support Robust Security | 24 |
| 5. Operational Considerations | 25 |
| 5.1. Schedule Domain Consistency | 25 |
| 5.2. Incremental Deployment | 26 |
| 6. Security Considerations | 27 |
| 6.1. Denial-of-Service (DoS) Attack | 27 |
| 6.2. Traffic Analysis and Path Prediction | 28 |
| 6.3. Activity Identification and Privacy | 28 |
| 6.4. Spoofing and Manipulation of Time Information | 28 |
| 6.5. Replay Attacks on Time-Sensitive Data | 29 |
| 6.6. Compromised Time Sources | 29 |
| 7. IANA Considerations | 29 |
| Acknowledgments | 29 |
| Contributors | 30 |
| References | 30 |
| Normative References | 30 |
| Informative References | 30 |
| Authors' Addresses | 32 |

1. Introduction

This document is an informational specification meant to inform the design and implementation of systems that manage time-variant routing (TVR) information, and to characterize those systems using design aspects understandable to network operators. The terms discussed in this document are intentionally general and are intended to be tailored for specifics of those individual TVR systems.

The motivation for this work is explained in the TVR Use Cases document [RFC9657], which justifies why there is value in having some form of time-variance in a system. This document discusses technical detail of aspects that designers and operators could adopt and considerations for when (or when not) to need to incorporate each of these aspects in a system design.

This document starts with an overview of TVR networks and aspects of their time variance in Section 2 and elaborates on TVR use cases in Section 3. Requirements on the design of TVR systems are then categorized and summarized in Section 4 with operational considerations for those systems in Section 5, and security considerations in Section 6.

1.1. Conventions and Definitions

Specific terms used within this document are as follows:

Model: The universe being modeled, which defines a parameter state space.

Entity: A single separable item within the model. Each entity has a stable identity which is time-invariant.

Property: A single attribute of an entity which is used to parameterize that entity. The notion of a property is not time-variant, the property always exists within an entity but its value may be time-variant.

Property Value: The specific value of a property, both as a planned state within the schedule timeline and as a realized state in wall-clock time.

Schedule: The method of parameterizing time-variance intrinsic to a time-variant model. The parameters of a schedule are within the state space of the model.

Schedule Time: An idealized timeline within a time-variant model

over which entities and property values may change without a difference of state in the model itself. The notion of schedule time is intrinsic to the model.

Wall-Clock Time: The true timeline, measured in some time scale by some local ticker. The notion of wall-clock time is extrinsic to the model; even non-time-variant models allow for changes over wall-clock time, just as different model states rather than a change within the model itself.

Time Instant: A single instant of time, consistent with the concepts of date-time in [RFC3339].

Timeline: A discrete or continuous sequence of time defined over a specific time datum.

Time Horizon: A bounded interval of time used to limit the applicability of a schedule or timeline, consistent with the concepts of period in [RFC3339].

Subsequent: A time instant which is later in a timeline than some reference time instant.

Snapshot: A way to transform a model with scheduled time-variance into one that is time-invariant and applies only to a single time instant. That instant need not be the current wall-clock time. The term snapshot can be used as a verb, to mean the transformation itself, or as a noun, to mean the output of the transformation.

Schedule Domain: A set of time-variant entities that are expected to be configured and operated jointly on the same timeline with a bounded synchronization of execution in wall-clock time.

Orchestrator: The subsystem of a managing device which centralizes control of a network and applies policy to manage a network. A Path Computation Element (PCE) [RFC4655] is an example of an Orchestrator.

Manager: The subsystem in a managing device which operates a management protocol to control an Agent.

Agent: The subsystem in a managed device which operates a management protocol to be controlled by a Manager.

Management Protocol: The mechanism used to exchange data between Managing and Managed devices, governed by a data model shared by the two devices.

(Routing) Application: The subsystem of a managed device which performs the functions of a routing protocol and/or algorithm.

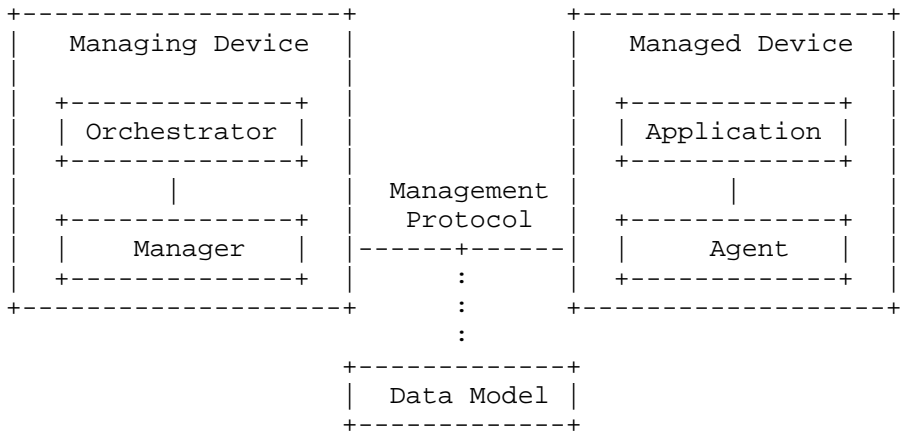


Figure 1: Management Entities

As a concrete example for the use of Figure 1 when applied to existing IETF standards, the management protocol could be NETCONF [RFC6241] which would be governed by a set of YANG data models [RFC7950] known to the Manager and its Orchestrator and implemented on the Agent and its Application. In this example, if the system uses all `_extrinsic_schedules_` the data model does not need to be time-variant and all schedule execution would occur within the Orchestrator. If the system uses any `_intrinsic_schedules_` then those need to be present within the data model, which would be communicated to and executed within the Application.

2. Overview of Time-Variant Networks

Existing Internet routing techniques maintain end-to-end connected paths across a network. Routing mechanisms exist to recover connectivity and resume normal traffic forwarding as the topology changes. Occasionally, optimization of routes may also be requested, especially post-topology changes due to disruptive events. However, there are a growing number of use cases where changes to the routing topology are an expected part of network operations. In these scenarios, the pre-planned loss and restoration of an adjacency, or formation of an alternate adjacency, should be seen as a non-disruptive event.

TVR refers to calculating a path or subpath through a network where the time of message transmission (or receipt) is part of the overall route computation. Therefore, a TVR computation might produce

different results depending on the time a calculation is performed without other detectable changes to the network topology or other cost functions associated with the route.

This section is organized into the following: Section 2.1 includes some basic definitions for when and how schedules would relate to a general data model, Section 2.2 discusses the temporal aspects within a schedule itself, Section 2.3 explains the entities of an IETF network model expected to benefit from a schedule, Section 2.4 discusses complex routing behaviors enabled by a network model with intrinsic schedules, and Section 2.5 discusses considerations needed when schedules are present in a data model.

2.1. Resource Scheduling

Planned resource scheduling is essential for various scenarios, including networks with mobile entities such as unmanned aerial vehicles and orbiting satellite constellations [RFC9657]. In these scenarios, links are lost and re-established as a function of the mobility of the platforms. Furthermore, link activity might be restricted to certain times of the day in networks without reliable access to power, such as networks harvesting energy from tidal, wind, and solar resources. Similarly, network traffic might be planned around energy costs or expected user data volumes in networks prioritising green computing and energy efficiency over data rate.

2.1.1. Schedule Domains

The concept of a schedule domain is to allow partitioning the universe of managed entities into separate sets of entities, each with independent timelines having independent schedule execution (Section 2.1.4) and likely independent schedule generation (Section 2.1.3). Within each domain, all schedules need to use the same timeline, need synchronized execution, and joint schedule generation.

Two extremes of how a system can organize schedule domains are:

Universal Domain: This extreme is to combine all scheduled state together into a single domain and single timeline. For cases where all nodes in a network have time-varying properties that affect their topological neighbors, all of the nodes need to be scheduled in the same domain to avoid misaligned configuration between devices. The orchestration and management burden in this case is the need to consider all schedules jointly and the operational burden is the need for synchronization of schedule execution across managed devices.

Per-Device Domains: This extreme is to consider each managed device as a separate scheduled domain, where property changes between devices need not be synchronized. This simplifies aspects of schedule execution but would likely rely on control plane communication between devices to avoid mismatch between the actual configuration of any node and how it is assumed to be configured by other nodes. The orchestration of schedules within a device in this case would likely be based on device-local needs such as for power control.

In cases where there does not need to be tight synchronization between some schedules, they can be managed in separate domains and effectively form separate timelines where the schedule time in one domain has little to no relation with any other domain.

A system design can choose to make some "edge" nodes or some properties of edge nodes intentionally time-invariant in order to form a logical boundary around schedule domains or possibly aligning schedule domains with routing domains so that schedule edge nodes correspond with routing edge nodes. This would allow routing control protocols to be used for online negotiation at domain boundaries.

2.1.2. Schedule Visibility

Because scheduled time-variance is not a part of existing routing algorithms and managed data models, not all routing applications will be made to handle schedules as part of the routing parameters intrinsically.

Two extremes of schedules being associated with routing data are:

Intrinsic Schedule: In this situation, the schedule is an intrinsic part of the managed data model which is visible to the routing application and used as part of the routing algorithms. When the schedule is intrinsic, there is not necessarily the notion of a schedule being "executed" as a single activity in wall-clock time because the time-varying parameters are ingested as part of the routing algorithm functioning (see Section 2.4) when routing is needed.

Extrinsic Schedule: In this situation, the schedule is not part of the managed data model for the Managed Device but maintained within the Orchestrator; the routing application only sees the effects of changes in routing parameters as the schedule is executed (in wall-clock time) by the Orchestrator.

There is also the possibility of an intermediate situation where the schedule is still part of the managed data model but is visible only to, and executed in wall-clock time by, the management Agent. This allows a more distributed use of scheduled data than centralizing its processing in an Orchestrator.

2.1.3. Generation Locality

The generation of a scheduled data model depends on collecting source data (which likely has some temporal information in it to begin with), choosing a time horizon to schedule within, and then processing the source data into an overall schedule.

Two extremes for locality of schedule generation are:

Centralized Generation: In this situation, all schedule generation is centralized within a network Orchestrator and changes are sent to routing applications in wall-clock time via a management interface. Even though the generation of the schedule is centralized, both the schedule visibility (within the data model) and the locality of how the schedule is executed are unconstrained.

For example, a schedule could be generated in a central orchestrator synchronized to all managed devices which then execute the schedule in a distributed manner.

Distributed Generation: This situation corresponds with the intrinsic or intermediate schedule visibility. Where a schedule (with a potentially limited time horizon from what is known at the orchestrator) is part of the managed data which is distributed to managed devices to be handled either by the Agent or by the routing Application itself.

2.1.4. Execution Locality

Depending on the visibility of schedules within a data model (see Section 2.1.2) there are different options for where the schedule may be executed, and ultimately influence a time-varying configuration on a managed device.

Two extremes for locality of schedule execution are:

Centralized Execution: In this situation, all schedule execution is

centralized within a network Orchestrator and changes are sent to routing applications in wall-clock time via a management interface. This situation can apply to any type of schedule visibility, but only to centralized generation because the full scheduled data model needs to be available to the entity performing the execution.

Distributed Execution: In this situation, schedules are executed on each managed device independently but based on synchronized clocks. This situation corresponds with the Intrinsic or intermediate schedule visibility, where a schedule (with a potentially limited time horizon from what is known at the Orchestrator) is part of the managed data which is distributed to managed devices to be handled either by the Agent or by the routing Application itself.

When schedules are distributed to the managed devices, it necessarily increases the amount of data that the managing device needs to synchronize across the network. The ratio of increased size can be mitigated by only distributing a limited time horizon to each device within a sliding window that moves forward in non-real-time.

When schedules are both generated and executed centrally, there is a consistency risk between different managed devices because if one device fails to be reconfigured in wall-clock time its configuration will no longer align with the other devices which are supposed to all operate on the same schedule. To recover from this kind of situation, either reattempt to configure the misaligned device may be made to bring it back into alignment with the other devices or the other devices' configurations must be rolled-back into consistency which will then cause all the devices to be off-schedule.

When schedules are executed on each device, there is a risk that clocks on different devices become de-synchronized beyond the time precision required of the schedule. Because real-time clocks are necessary for more than just schedule execution, and because accurate and precise time sources exist outside of network time (e.g., GPS time) this risk can be made to have a low probability.

With distributed execution there is also a risk that a Manager loses connectivity with the managed device and the device eventually runs out of time horizon in the schedule which is known to it. This risk can be mitigated by trading between the size and the horizon end-time of schedules distributed to managed devices. This trade can be different for different devices, where some well-connected devices operate closer to just-in-time with short horizons while other devices can be given a longer horizon to allow it to execute in the absence of near-continuous Manager connectivity.

One possible combination of these options is depicted in Figure 2, where inputs are collected in a centralized schedule generator, the schedule is executed on that centralized entity by taking a snapshot (periodically or as-needed when model state changes over schedule time) and distributing the time-invariant snapshot configuration.

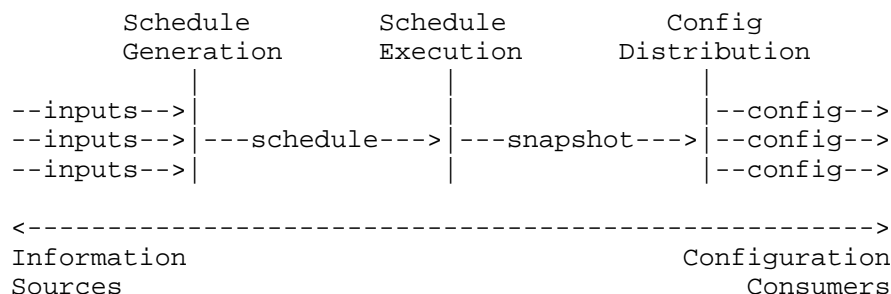


Figure 2: Centralized Generation with Centralized Execution

An alternative combination is depicted in Figure 3, where inputs are also collected in and a schedule generated by a centralized entity, but in this alternative the scheduled data model (or some filtered time horizon of it) is distributed to the managed devices to be executed independently on each device.

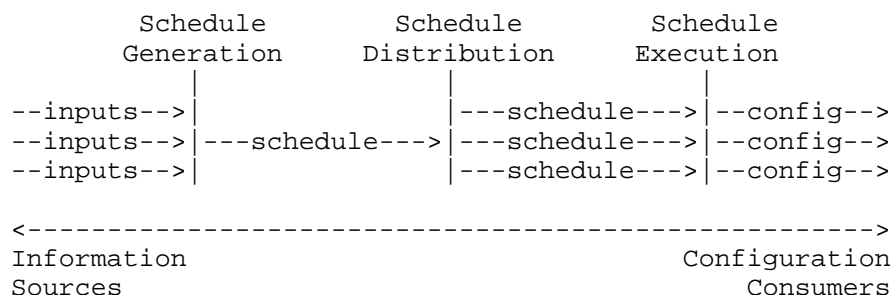


Figure 3: Centralized Generation with Distributed Execution

2.1.5. Configuration and Operational State

Most of the discussion in this document treats scheduling as the means for influencing when configuration on some managed device is planned to be updated. But as explained in [X.731] devices are expected to have an operational state alongside many administrative states being configured. For example, a known delay between enabling the modem supporting a termination point or link and the modem actually being usable for sending traffic.

Strategies for modeling time margins around changes of configuration are discussed in Section 2.2.4. Even when time margins are taken into account, the schedules are still being applied to cascading subsequent changes of administrative state within configurations. Those changes can motivate subsequent changes in operational states. While the administrative changes follow the schedule times, the operational states could be effective at different times across devices (_e.g._, because of different implementations or other device-specific reasons).

2.2. General Temporality

This section covers different aspects of how temporality applies to any potential TVR data model or TVR augmentation of a time-invariant data model. Each aspect is roughly independent and informs how a model can choose to include temporality in its parameter state space.

Each of these aspects can be different across different schedule domains (Section 2.1.1), but are expected to be consistent within a single schedule domain. Also, just because an entire model or domain allows high granularity (Section 2.2.1) or high precision (Section 2.2.3) does not mean that every single entity needs to make use of those aspects (or even that every entity needs to have time-variance at all). It is perfectly valid for some entities to have time-variance and others to have none.

2.2.1. Scope of Time-Variability

One aspect of any time-variant data model is the scope of what may be time-variable. Two extremes of this aspect are:

- * A model that is entirely time-invariant, where time exists conceptually but has no impact on any of the model's entities.
- * A model in which every entity has some kind of schedule applied.

It is expected that an application of time-variability to real world data models will keep some entities within the model time-invariant and allow scheduling of other, specific entities.

Another aspect of any time-variant data model is the granularity of state to which a schedule can be applied. Two extremes of this aspect are:

- * A model where one single schedule applies to the entire universe (_i.e._ indicating when the time-variant entities are valid or invalid).
- * A model where every property of every entity can be scheduled independently. This is the temporality model of [AIXM].

It is expected that the use of time-variability in data models will fit within these extremes. One possibility is to apply a schedule to each entity indicating when that entire entity is valid or invalid. Another possibility is to apply a schedule to groups of properties within an entity (while leaving other properties time-invariant).

2.2.2. Time Horizon

In an idealized model the schedules will apply indefinitely far in the past and the future, but in a realizable data model with both processing and storage limitations there will need to be a time horizon within which the model applies and outside of which the model has no meaning. In some cases this horizon will be intrinsic to the data model itself, with an explicit model parameter indicating the horizon. In other cases the data model may allow indefinitely-large schedules but the processing of the schedule timeline is bounded to limit resource needs.

One possible rationale for separating schedule domains (Section 2.1.1) is the duration of the time horizon needed for entities in each domain.

2.2.3. Time Precision and Accuracy

Different time-variant models will require different granularities of planning time, either because of limitations or assumptions about wall-clock time or because of requirements within the modeled domain. It is up to specific models to define the precision of time values and the required accuracy and precision of wall-clocks which execute the schedules.

One possible rationale for separating schedule domains (Section 2.1.1) is the level of time precision or accuracy of execution time able to be upheld across entities in each domain.

2.2.4. Time Synchronization and Margin

Any schedule execution and device configuration (see locality options in Section 2.1.4) will necessarily have some misalignment in the synchronization of time across all devices operating in the same timeline. This misalignment is hopefully bounded by design and able to be characterized statistically.

It is important for the activity of schedule generation (see Section 2.1.3) to take the misalignment into account as some form of margin around the instants of scheduled change. The exact form that this margin would take depends on the specific time-varying properties.

Another source of time margin in a time-varying system can be due to a desire to model the time delay between changing the administrative state of some subsystem and a subsequent change to its operational state as a consequence.

Regardless of the reason for a schedule margin being accounted for, it is critical that the margin is not double-counted by different activities in the schedule processing chain.

2.2.5. Validity in a Schedule

Within a single schedule over its timeline there will likely be a need to have multiple discrete intervals of validity over absolute schedule time. The time instants at which a schedule is invalid indicate an undefined property value, so it is important for a model to be able to accommodate multiple schedules as necessary to ensure that some properties can have values at all times.

A model which restricts itself to a single interval of validity could run into difficulties over a long enough time horizon and would need to resort to having multiple model entities represent the same modeled "thing" which can lead to confusion and inefficiency.

2.2.6. Periodicity in a Schedule

Separate from the concept of intervals of validity in absolute schedule time, there can be a need to model repetitive states in a concise way. One way to model a periodic change of state is to combine a set of absolute time intervals with a periodic parameterization (duration valid and duration invalid); this is the model of [AIXM].

A model which does not include the notion of periodicity within a schedule could be used in situations where discrete intervals of validity are needed to handle periodic state changes which is neither storage nor processing efficient. Periodicity can also be seen as unnecessary when the time horizon will always be small enough compared to any schedule time period that only one repetition is ever seen within the horizon in one state.

2.2.7. Continuity in a Schedule

A schedule which includes a sequence of time intervals needs to ensure that the interpretation of those intervals in the schedule timeline does not leave any "gaps" at the interval boundaries. For that reason, it is important that the model uses half-open intervals of time so that time-adjacent intervals leave no gap. In keeping with the terminology of [RFC3339], intervals are bounded by their "start" and "end" instants. It is suggested that any time-varying model use schedules with intervals closed on their start time and open on their end time. This behavior lends to the interpretation, in the schedule timeline, that the scheduled state takes effect at an interval's start and continues until the subsequent state.

2.2.8. Time-Overlap and Priority

In an ideal situation a model would be guaranteed by design to contain only contiguous and non-overlapping schedules for each time-variant scope. In a realized model this kind of invariant might not be enforceable or might lead to overly complex schedule structures. One way a model can handle this is to establish a concept of schedule priority, where some intervals of the schedule timeline contain overlapping schedules for the same properties and only the highest-priority schedule applies. When priorities are allowed by a model, it enables the concept of an "overlay" where a long-duration state can be temporarily (in schedule time) superseded by a short-duration state.

2.2.9. Property Value Interpolation

When a schedule is applied to an entity in a way which is more granular (Section 2.2.1) than just indicating when that whole entity is valid or invalid, the model needs to consider how individual properties are to be treated between scheduled instants. Some of the possible behaviors are:

Zero-order hold: From the instant a scheduled value applies to a property until the subsequent-in-schedule-time value supersedes it. This is simple from a logical standpoint, but discontinuities in the value over schedule time could cause issues with the model itself. For some models, though, the constant values between change instants are actually beneficial by allowing the entire timeline to be compressed into a sequence of discrete state-change instants. This is the behavior implied in models such as [AIXM].

Linear interpolation: At the instants of time defined in the schedule the property takes the exact values, but between those instants the property is interpolated linearly over time. This results in a state that is continuous over time, which is beneficial for some kinds of model but also means that there is no simple discrete sequence of states.

Higher-order or spline interpolation: Higher order interpolations can result in properties that vary over schedule time in ways that are more or less beneficial to different types of models.

Regardless of the types of interpolation used, a model can choose to apply interpolation globally or per-property. Since different properties represent different physical or logical metrics of a network it is expected that different types of interpolation will be needed for different represented quantities.

2.2.10. Changes to Model State

Separate from how a time-variant model can contain a schedule timeline within the model state, a model design will need to consider how changes to the model state itself (over wall-clock time) are handled. This aspect is actually not specific to a time-variant model but is important to consider in this context.

Two extremes of this aspect are:

- * A model which can only be changed wholesale, superseded by an entire new model state. This is easy to keep consistent but has inefficiencies of storage and transport if the model state is to be shared or exchanged between real entities.

- * A model which has an intrinsic notion of fine-grained superseding changes, possibly scoped to individual entities, individual schedules, or more complex groupings.

2.3. Topologies

The primary entities of a topological network model, as realized in the IETF [RFC8345] and similar predecessors, are `_nodes_` and `_unidirectional_links_`, with a secondary entity representing the `_termination point_` for each side of a link at a node. Following the concepts described in Section 2.1 these are the entities to which an intrinsic schedule can be applied. Since TVR is focused on the routing aspect of scheduled systems, relating schedules to entities in a network model used for routing is meant to give concrete guidance about where there is value to put a schedule in a TVR system.

2.3.1. Nodes

When a schedule is applied to a node the granularity could at least be at the individual node. In cases where the properties of a node have time-variable values the model may define an interpolation method, either globally or per-property.

A node is just a named entity in Layer 3 [RFC8346] and Layer 2 [RFC8944] topologies. Schedules on a node could be used to indicate the validity of the entire node or changing properties of that entity. When a schedule indicates that a node is not valid for a schedule time instant, that validity could apply to all of its termination points and links as well. This logic allows a schedule to represent, for example, the expected power-on state of a node at a specific layer.

2.3.2. Termination Points

When a schedule is applied to a termination point the granularity should at least be at the individual entity. In cases where the properties of a termination point have time-variable values the model may define an interpolation method, either globally or per-property.

A termination point is associated with an IP address in Layer 3 [RFC8346] and a MAC address in Layer 2 [RFC8944] topologies. Schedules on a termination point could be used to indicate the validity of the layer-2/3 interface represented by the entity or changing properties of that entity. When a schedule indicates that a termination point is not valid for a schedule time instant, that validity may apply to all of its links as well. This logic allows a schedule to represent, for example, the expected power-on or administrative-enabled state of an attached network interface card (NIC) or virtual private network (VPN) endpoint.

2.3.3. Links

When a schedule is applied to a link the granularity should at least be at the individual link. In cases where the properties of a link have time-variable values the model should define an interpolation method, either globally or per-property.

A link is associated with link metric properties in Layer 3 [RFC8346] and Layer 2 [RFC8944] topologies. Schedules on a link should be used to indicate the validity of the entire link or changing properties of that entity. When a schedule indicates that a link is not valid for a schedule time instant, that validity should not apply to its termination points and nodes. This logic allows a schedule to represent, for example, the expected connectivity state, data throughput/rate, and latency/delay of a link.

2.3.4. Network Layering

When a schedule indicates that an entity is not valid for a schedule time instant, that validity should not apply to any of its associated overlay or underlay network entities. The effects of scheduled administrative disabling or enabling of an entity at one layer do not imply a change in administrative enabled state at any other layer. Likewise, the assigning of an address property at one layer does not imply the presence or absence of an address assignment at that same time instant for any other layer.

2.4. Routing Strategies

Traditional network routing techniques typically use link bandwidth and delay for path calculation, and do not consider time-based factors. TVR should be capable of improving network performance and reliability in environments where entities liveness and link availability is a time-based consideration, with various factors, including power availability, interface line of sight or expected demand.

However, even if some adjacency failures are predictable, others are not, including link failures and entity outages. Therefore, any new technique or routing protocol extension for TVR environments must be capable of handling planned and unexpected resource losses or other changes.

TVR introduces a scenario of calculating a path, or sub-path within a network, taking into account the timing of message transmission or receipt as an integral part of the overall route computation.

Furthermore, synchronization of network time across TVR-capable entities is critical.

Three scenarios are currently considered when computing TVR-enabled paths and described in the following subsections.

2.4.1. Centralized

The network entities will receive the time variable information and traffic forwarding rules directly from a logically centralized source, an Orchestrator or network controller. The time-variable data may then be processed locally by the entity entered into the scheduled routing table and specific forwarding rules applied.

Furthermore, a centralized approach could also be used to extend existing tunnel and path delivery mechanisms and protocols to distribute traffic forwarding rules along with time-variable information. However, in certain environments, a logically centralized source may lose connectivity with network entities (as described in Section 2.1.4), preventing timely delivery of traffic forwarding rules. To mitigate this risk, the time horizon for time-variable information should be extended accordingly.

2.4.2. Distributed

Network entities may participate in a routing scheme where time variable information is propagated through the network via capability and variability advertisements. This could be achieved using extensions to existing routing schemes and techniques so that link, adjacency, cost, and schedule may be considered when making forwarding decisions for per-hop packets or calculating traffic engineered end-to-end paths. It should be noted that schedule distribution and entity computation latency may exist in some network environments.

In some environments, scheduling information may be distributed through a management plane mechanism, such as NETCONF [RFC6241] or gNMI (gRPC Network Management Interface) [gNMI], instead of the routing scheme.

2.4.3. Hybrid

In this scenario, mixed-entity TVR capability exists. Some entities will require a schedule provided by a centralized source, and others will be capable of advertising and learning scheduled information via a distributed mechanism.

This scenario presents time and schedule synchronization and source verification challenges and will require further study, but are out of scope for this document.

2.4.4. Constraints

Time-variant network constraints may be based on dynamic factors that will influence how the network is managed and how network resources are scheduled. These constraints are influenced by real-time data and can vary significantly depending on multiple factors. By considering time-variant constraints, network operators can enhance the efficiency, reliability, and performance of telecom networks. The main factors influencing these constraints include:

1. **Predicted Traffic Demand:** Network usage patterns fluctuate throughout the day, with peak times typically occurring during business hours and in the evening. Predicting these patterns accurately allows for proactive resource allocation, ensuring that sufficient bandwidth is available during high-demand periods without over-provisioning during low-demand times.
2. **Energy Efficiency:** The energy consumption of network equipment can be optimized based on the current load. By scheduling resources and adjusting power levels or shutting down underutilized equipment, telecom networks can significantly reduce energy costs and carbon footprints, contributing to sustainability goals.
3. **Weather Conditions:** Weather can impact network performance, especially for wireless and satellite communications. Adverse weather conditions such as heavy rain, snow, or extreme temperatures can degrade signal quality. Incorporating predicted and real-time weather data into network management strategies can help in adjusting transmission power, rerouting traffic, or preemptively switching to more resilient pathways.
4. **Network Maintenance and Upgrades:** Scheduled maintenance or unexpected faults can introduce temporary constraints. By planning maintenance activities during off-peak hours and having real-time monitoring systems to quickly detect and address faults, network downtime can be minimized.

2.5. Integrity Considerations

Time-variant network relies on accurate and timely dissemination of time-variant routing and forwarding information. However, the presence of malicious or unintended divergent information introduces risks that can impact network stability and operational correctness. An adversary could manipulate scheduled routing updates to introduce black holes, persistent loops, or denial-of-service conditions by injecting false time-sensitive state changes. Even in non-malicious scenarios, incorrect or misaligned scheduling or misconfiguration, or time de-synchronization, may lead to unintended forwarding behavior, potentially degrading performance or causing service disruptions.

To mitigate these risks, TVR solution mechanisms should incorporate integrity validation and trust enforcement to ensure the correctness and authenticity of time-sensitive routing updates. This may include cryptographic techniques to verify the source and integrity of schedule updates, consistency checks against expected network state, and mechanisms to detect and reject anomalous scheduling data. Additionally, fallback strategies should be considered to allow continued operation in cases where unexpected or inconsistent information is detected.

Specific security considerations are discussed in the Section 6. later in this document.

3. Time-Variant Use Case Requirements

Several TVR use cases have been identified and discussed in [RFC9657]. This section provides further detail on specific requirements to meet use case needs.

3.1. Resource Preservation Use Case

This use case is about managed devices being reactive to sensed conditions, but also providing feedback to an Orchestrator to allow coarse schedules of expected resource availability. Its requirements include:

Temporality tailored to system dynamics: Because managed devices are either powered-off or severely degraded in performance when preserving resources, the schedules governing expected topology must be of sufficient precision and synchronization to capture the dynamics of the managed devices.

Parameterization of periodicity: Resource availability based on

diurnal activity on Earth fits well into a periodicity based strictly on time-of-day. But when resources are available based on other natural phenomena (e.g., orbital periods) the schedules must have a periodicity which is parameterized in such a way that allows matching the resource dynamics (e.g., repeating subsequent intervals of durations of seconds available and seconds unavailable).

Time horizon tailored to uncertainty: Even when periodicity is well understood, characterized, and parameterized in a data model, there must be allowance for uncertainty of expected resource availability. This operates both in the sense of having a large enough time horizon to enable a device to "ride out" times of low resources without needing schedule updates, as well as the horizon being limited to avoid large schedules which far exceed the point where actual system state will diverge from the model state.

3.2. Operating Efficiency Use Case

This use case is about scheduling resources proactively to improve efficiencies. Its requirements include:

Distribution of Predicted Topology-change: The predicted topology-change information may include the valid time, invalid time, link costs at different times, and change periods.

Topology Changes: The predicted topology-change information may change due to forecasted or unforecasted changes. The managing entity should be capable of providing a partial or full topology update as often as needed.

Minimum Route Recalculation Interval and Threshold: Although some cases may assume that the cost persists for a sufficient amount of time, considering that each route contains multiple links, the change frequency of the path may be much higher than the cost. In this case, the minimum recalculation interval or cost change threshold is needed to determine when a route recalculation is required. Of course, scheduled topology connection changes must be considered when path calculation is required.

3.3. Dynamic Reachability Use Case

This use case is about geometric and kinematic constraints of mobile devices influencing their ability to establish or maintain links to neighbors. Its requirements include:

Pairwise consideration of synchronization and margins: When

scheduling links, the execution clock synchronization of its two endpoints as well as any margin needed on each of those endpoints must be considered when generating schedules for those links. Additionally, schedules on long-distance (*_i.e._*, interplanetary scale) links must consider the effects of light-speed delays for distribution and execution.

Schedule independence from external conditions: An important concept of distributed schedule execution in TVR is that a consistent shared timeline and a wall-clock ticker across managed devices is the unified mechanism to synchronize state across devices. Schedule entries should not depend on externally sensed conditions such as location, orientation, or other geometric or kinematic properties. Changes arising from such external conditions that are not captured in the schedules are outside the scope of TVR and are handled reactively by the network. Schedule-based TVR and condition-based reactive mechanisms can coexist within the same network and within its devices.

4. Requirements Summary

4.1. Support the Identification and Advertisement of Entity Property Changes

In Time-Variant Routing, scheduling of available entity resources is expected. In practical situations, however, the properties of entities can be converted back and forth between Time-Variant and Non-Time-Variant nodes.

An entity must support the identification and advertisement of non-scheduled property changes.

Besides, if there are abnormal changes in the system, it is necessary to advertise them through the existing routing protocols in time to achieve the stability of Time-Variant Routing and avoid redundant advertisements. For example, an entity in the system is suddenly damaged due to external factors. Changes in entity state outside of a schedule are communicated to other entities in a network through existing routing protocol mechanism, where they exist.

A Manager should provide an advertisement methodology for responding to abnormal changes in the system.

4.2. Support Proxy Advertisement

Proxies can help to improve the efficiency of the network. There are some entities in the network that do not have routing functions. When their properties change, they are unable to notify other entities in the network. Proxy nodes can help nodes without routing functions to advertise information, thus improving the efficiency of the network. Therefore,

Systems must support proxy entities to help non-routing nodes implement information advertisement.

4.3. Support Identification and Classification of Node Properties

The entity properties of the network may change as described in 3.1. If the system cannot timely identify and classify in a processing manner after the entity properties change, it will lead to suboptimal routing decisions. Therefore,

Systems must provide a discovery and resolving methodology for the identification and classification of entity schedule changes.

4.4. Support System Schedule and Time Interval Changes

The system's schedule may change, requiring entity configuration updates rather than being fixed and unmodifiable. Additionally, time-variant intervals in the system may also vary. Therefore,

Systems must support system schedule changes.

Systems must support time interval changes.

4.5. Support Appropriate Time Accuracy

The accuracy of the time cannot be too large or too small; otherwise, convergence may not be possible. Therefore,

Systems must support appropriate time tolerance.

4.6. Support Robust Security

Implementations must address security risks associated with time-variant information to ensure the reliability and integrity of scheduled network operations. The following security-related requirements should be considered,

Integrity Protection: Mechanisms must be in place to ensure that

time-sensitive routing updates are protected from unauthorized modification.

Authentication and Authorization: Entities generating or modifying TVR schedules must be authenticated, and only authorized entities should be permitted to inject, update, or override scheduled routing information.

Resilience Against Malicious or Erroneous Inputs: A TVR network must be resilient against the injection of incorrect or maliciously crafted scheduling information.

Time Synchronization Robustness: Since TVR relies on time-sensitive operations, it must ensure the trustworthiness of external time sources. Protection against time-based attacks, such as replay attacks or clock manipulation, should also be considered.

Rollback and Recovery: In the event of conflicting, missing, or compromised time-variant routing data, TVR implementations should include fallback mechanisms to maintain network stability.

By integrating these security requirements into TVR implementations, networks can mitigate risks associated with malicious actors, misconfigurations, or unintended disruptions, ensuring the robustness of time-sensitive routing decisions. Specific security scenarios and negation and mitigation methods are discussed in Section 6.

5. Operational Considerations

Introducing time-variance to network operations and management in any capacity adds complexity to those areas of system design and implementations. This section discusses considerations for those areas in the spirit of [RFC5706] but without concrete details of a specific TVR system design.

5.1. Schedule Domain Consistency

As explained in Section 2.1.1, the purpose of a schedule domain is to organize managed devices based on their time-variant needs and capabilities. The choice of which devices to include in a domain is subjective, but should take into consideration the schedule-awareness capabilities of the devices and temporal sensitivities of their configurations.

For example, including in a single domain devices which can handle data models with intrinsic schedules and devices which cannot will increase the burden on the network orchestrator to validate the joint configurations of those devices. Segmenting into different schedule

domains would allow for a more simple validation of the time-invariant device configurations and a more narrow but complex validation of the time-variant device configurations.

Another important consideration for schedule domains is the required wall-clock time precision and accuracy of devices in each domain, as explained in Section 2.2.3. A single domain which includes devices needing only coarse time precision as well as those needing tight precision can add unnecessary burden to planning and validation of schedules for the coarse-precision devices.

Similarly, the required time synchronization for devices in each domain, as explained in Section 2.2.4, affects the amount of and types of analysis that a network orchestrator needs to validate configurations of those devices. A single domain which includes some devices needing only loose synchronization and some with very tight requirements adds burden to planning and validation of schedules.

5.2. Incremental Deployment

There is an expectation that the intentionally simplified view of Figure 1 would actually contain a large number of separate but possibly inter-related data models being managed for even a modestly complex managed device. And that figure shows only a single managed device while a real network is expected to contain a large number of managed devices, possibly with a diverse set of management protocols serving different sets of devices, but under the control of a central orchestrator. Because of these expectations, the introduction of TVR into an existing management ecosystem is meant to be able to be deployed incrementally, possibly along different aspects in different increments.

Within the aspect of schedule domains (Section 2.1.1) it is possible to start with a single completely time-invariant domain, add TVR as a single time-variant domain covering a portion of devices most in need of scheduled behavior. From there the operator can either grow that time-variant domain to cover more devices or add other time-variant domains to suit the operator's needs over deployment increments.

Within the aspect of schedule visibility (Section 2.1.2) it is possible to deploy TVR first as extrinsic schedules known only to the network orchestrator and executed centrally there. From there the operator can transition a portion of those extrinsic schedules to intrinsic schedules, which requires support for (some form of) schedule execution on the managed devices. As needs change or device support is updated, the scope of intrinsic schedules can be grown or adjusted to suit over deployment increments.

Within the aspect of generation locality (Section 2.1.3) a time-variant domain can start out as a fully externally-controlled device and possibly expand to allow managed devices themselves to propose schedules based on locally-sensed conditions such as traffic periodicity or resource (e.g., power) availability. Managed devices might propose schedules which can then be simulated and verified by an orchestrator and augmented as intrinsic schedules back to those devices.

The complexity and scope of supported schedules can also be adjusted incrementally, starting with a time-variant domain that operates on very simple schedules with coarse-grained scope and short time horizons. From there, an operator can incrementally increase schedule complexity, make schedule scope more fine grained, or expand time horizons as device support is updated. These changes can all be mediated through schedule domains focused based on that device support.

6. Security Considerations

Using time-variant mechanisms introduces unique security vulnerabilities that must be carefully considered to ensure the integrity, availability, and confidentiality of the network. Networks relying on time-sensitive data for forwarding decisions are particularly susceptible to attacks that exploit temporal aspects and timing dependencies.

The following potential security considerations warrant detailed investigation as solutions are developed and deployed.

6.1. Denial-of-Service (DoS) Attack

Precisely coordinating time information across devices and routers is critical to maintaining network stability. Malicious actors could exploit this dependency by disrupting or manipulating the time synchronization process. For example, an attacker could intentionally delay or corrupt time signals exchanged within the network, leading to routing errors and widespread denial-of-service (DoS) attacks. In this scenario, routers and managed devices may fail to correctly determine the optimal paths, resulting in dropped packets, increased latency, or even complete service outages. Additionally, these attacks could be scaled to affect multiple devices simultaneously, further amplifying their impact. Given the critical nature of time in such networks, securing time synchronization mechanisms, such as Network Time Protocol (NTP) or Precision Time Protocol (PTP), is essential to mitigate these risks.

6.2. Traffic Analysis and Path Prediction

Time variant networks may involve frequent updates and adjustments to routing tables based on current and forecasted network conditions. If time information is not adequately protected, attackers could conduct traffic analysis to infer routing decisions, network load, or usage patterns. The schedule ability could enable attackers to launch highly targeted attacks, such as selectively overloading certain links or intercepting sensitive communications. Moreover, long-term analysis of time-variant network data could provide attackers with insights into the underlying structure of the network, enabling them to plan more sophisticated attacks. To counter these threats, it is vital to encrypt time-sensitive data and limit the exposure of time-related metadata to unauthorized entities.

6.3. Activity Identification and Privacy

In certain scenarios, precise time information exchanged within the network could be correlated with specific user or device behavior, inadvertently revealing private information. For instance, time scheduling decisions could be analyzed to determine when and where certain devices are active, allowing an attacker to infer user habits, locations, or preferences. This could pose significant privacy concerns, particularly in environments where sensitive personal or organizational data is transmitted. Furthermore, attackers could use this information to create detailed profiles of network users, which could be exploited for social engineering attacks, surveillance, or other malicious activities.

6.4. Spoofing and Manipulation of Time Information

The accuracy and integrity of time information are crucial for making correct routing decisions. If an attacker were to inject false or manipulated time data into the network, it could cause routers and devices to make incorrect decisions, potentially leading to traffic misrouting, network partitions, or inefficient use of resources. Such spoofing attacks could divert traffic through malicious nodes, enabling man-in-the-middle attacks, data interception, or unauthorized access to network resources. Furthermore, time manipulation could create persistent disruptions by continuously altering the perceived time, thereby forcing the network into a constant state of flux and instability. Robust authentication mechanisms for time sources and integrity checks on time-related messages are essential to defend against these types of attacks. Moreover, implementing redundancy in time synchronization (e.g., multiple time sources) can provide resilience against single points of failure.

6.5. Replay Attacks on Time-Sensitive Data

Replay Attacks on Time-Sensitive Data: Time variant network data and schedule updates may be susceptible to replay attacks, where a malicious actor intercepts and retransmits valid time-based data at a later time. This could cause network devices to act on outdated information, leading to inconsistent routing decisions, misaligned schedules, or security gaps. In particular, attackers could exploit replay attacks to force devices into outdated configurations or interfere with the synchronization of schedules across the network. To prevent this type of attack, it is important to use a messaging protocol for time-variant schedules that mitigates such attacks while ensuring the validity and timeliness of received information.

6.6. Compromised Time Sources

Compromised Time Sources: The reliance on external time sources for synchronization purposes presents a potential attack surface for time-variant networks. If a trusted time source, such as a GPS signal or an NTP server, is compromised, the attacker could feed erroneous time information to the entire network, disrupting its operation. Such an attack could lead to cascading failures as devices attempt to synchronize with the compromised source, ultimately resulting in incorrect routing decisions or even the collapse of the network. To address this, network operators should implement multiple, redundant time sources and regularly verify the integrity of these sources. In addition, alerting mechanisms should be in place to detect significant deviations in time data that could indicate an attack.

7. IANA Considerations

This document has no IANA actions.

Acknowledgments

This work has benefited from the participation of the TVR working group and the discussions on the mailing list.

The authors would like to specifically thank Tony Li, Mark Blanchet, Alexander Petrescu, Ed Birrane, Jie Dong, Abdussalam Baryun and Joel Halpern

This work is partly supported by the UK Department for Science, Innovation and Technology under the Future Open Networks Research Challenge project TUDOR (Towards Ubiquitous 3D Open Resilient Network).

Contributors

The following authors contributed significantly to this document:

Jing Wang
China Mobile
China
Email: wangjingjc@chinamobile.com

Peng Liu
China Mobile
China
Email: liupengyjy@chinamobile.com

Zheng (Sandy) Zhang
ZTE Corporation
China
Email: zhang.zheng@zte.com.cn

Yuehua Wei
ZTE Corporation
China
Email: wei.yuehua@zte.com.cn

Charalampos (Haris) Rotsos
Lancaster University
United Kingdom
Email: c.rotsos@lancaster.ac.uk

References

Normative References

- [RFC9657] Birrane, III, E., Kuhn, N., Qu, Y., Taylor, R., and L. Zhang, "Time-Variant Routing (TVR) Use Cases", RFC 9657, DOI 10.17487/RFC9657, October 2024, <<https://www.rfc-editor.org/info/rfc9657>>.

Informative References

- [AIXM] EUROCONTROL and Federal Aviation Administration, "AIXM 5 Temporality Model", 15 September 2010, <https://aixm.aero/sites/aixm.aero/files/imce/AIXM51/aixm_temporality_1.0.pdf>.

- [gNMI] Borman, P., Hines, M., Lebsack, C., Morrow, C., Shaikh, A., Shakir, R., Li, W., and D. Loher, "gRPC Network Management Interface (gNMI)", Version 10.0, May 2023, <<https://www.openconfig.net/docs/gnmi/gnmi-specification/>>.
- [RFC3339] Klyne, G. and C. Newman, "Date and Time on the Internet: Timestamps", RFC 3339, DOI 10.17487/RFC3339, July 2002, <<https://www.rfc-editor.org/info/rfc3339>>.
- [RFC4655] Farrel, A., Vasseur, J.-P., and J. Ash, "A Path Computation Element (PCE)-Based Architecture", RFC 4655, DOI 10.17487/RFC4655, August 2006, <<https://www.rfc-editor.org/info/rfc4655>>.
- [RFC5706] Harrington, D., "Guidelines for Considering Operations and Management of New Protocols and Protocol Extensions", RFC 5706, DOI 10.17487/RFC5706, November 2009, <<https://www.rfc-editor.org/info/rfc5706>>.
- [RFC6241] Enns, R., Ed., Bjorklund, M., Ed., Schoenwaelder, J., Ed., and A. Bierman, Ed., "Network Configuration Protocol (NETCONF)", RFC 6241, DOI 10.17487/RFC6241, June 2011, <<https://www.rfc-editor.org/info/rfc6241>>.
- [RFC7950] Bjorklund, M., Ed., "The YANG 1.1 Data Modeling Language", RFC 7950, DOI 10.17487/RFC7950, August 2016, <<https://www.rfc-editor.org/info/rfc7950>>.
- [RFC8345] Clemm, A., Medved, J., Varga, R., Bahadur, N., Ananthakrishnan, H., and X. Liu, "A YANG Data Model for Network Topologies", RFC 8345, DOI 10.17487/RFC8345, March 2018, <<https://www.rfc-editor.org/info/rfc8345>>.
- [RFC8346] Clemm, A., Medved, J., Varga, R., Liu, X., Ananthakrishnan, H., and N. Bahadur, "A YANG Data Model for Layer 3 Topologies", RFC 8346, DOI 10.17487/RFC8346, March 2018, <<https://www.rfc-editor.org/info/rfc8346>>.
- [RFC8944] Dong, J., Wei, X., Wu, Q., Boucadair, M., and A. Liu, "A YANG Data Model for Layer 2 Network Topologies", RFC 8944, DOI 10.17487/RFC8944, November 2020, <<https://www.rfc-editor.org/info/rfc8944>>.
- [X.731] ITU, "Information Technology - Open Systems Interconnection - System Management: State Management Function", ITU-T X.731, 31 January 1993, <<https://www.itu.int/rec/T-REC-X.731>>.

Authors' Addresses

D. King
Lancaster University
Email: d.king@lancaster.ac.uk

L. M. Contreras
Telefonica
Email: luismiguel.contrerasmurillo@telefonica.com

B. Sipos
JHU/APL
Email: brian.sipos+ietf@gmail.com

L. Zhang
Huawei
Email: zhangli344@huawei.com