

Workgroup: opsawg
Internet-Draft:
draft-davis-opsawg-some-refinements-to-
rfc8345-01
Published: 10 January 2024
Intended Status: Informational
Expires: 13 July 2024
Authors: N. Davis O. Havel B. Claise
 Ciena Huawei Huawei
 Some Refinements to Network Topologies (RFC8345)

Abstract

This draft provides a brief analysis of the current unidirectional point-to-point approach to modeling of the link in RFC8345, highlights why this is not sufficient and makes a proposal to enhance RFC8345 YANG to support multipoint uni/bi links. The two alternative enhancement approaches proposed are backward compatible. The enhancement is such that it provides a uniform solution to modeling all links that could, over time, replace the current unidirectional point-to-point approach. The rationale for the change is based on many years of practical experience, including challenges using RFC8345 in actual solution development, and insight gained through other standardisation efforts and deployments.

About This Document

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

The latest revision of this draft can be found at <https://example.com/LATEST>. Status information for this document may be found at <https://datatracker.ietf.org/doc/draft-davis-opsawg-some-refinements-to-rfc8345/>.

Discussion of this document takes place on the WG Working Group mailing list (<mailto:WG@example.com>), which is archived at <https://example.com/WG>.

Source for this draft and an issue tracker can be found at <https://github.com/USER/REPO>.

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 13 July 2024.

Copyright Notice

Copyright (c) 2024 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](#)
- [2. Key terms](#)
- [3. Analysis of existing RFC8345](#)
 - [3.1. RFC8345 section 4.2 Base Network Topology Data Model](#)
 - [3.2. RFC8345 section 4.4.5 Cardinality and Directionality of Links](#)
 - [3.3. From the code: "list link {"](#)
 - [3.4. From the code: "container source {"](#)
 - [3.5. From the code: "container destination {"](#)
- [4. Enhancement approach](#)
 - [4.1. Observations](#)
 - [4.2. Essential Enhancement and usage](#)
- [5. Comparison of the existing point-to-point solution with this multipoint proposal](#)
- [6. Basic enhancement](#)
- [7. More sophisticated enhancement](#)
- [8. Further considerations](#)
 - [8.1. Termination direction](#)
 - [8.2. Specification of capability](#)
 - [8.3. Links between networks/subnetworks/domains](#)
 - [8.4. Richness of navigation](#)
 - [8.5. Clarification of relationship roles](#)

- [8.6. More than just links and termination points](#)
- [8.7. Layering and sub-layering](#)
- [9. Conclusion](#)
- [10. Implementation Status](#)
 - [10.1. Standards driven implementations](#)
 - [10.2. Proving the model](#)
- [11. Security Considerations](#)
- [12. IANA Considerations](#)
- [13. Informative References](#)
- [Appendix A. Appendix A](#)
 - [A.1. Examples of multipoint](#)
 - [A.1.1. Root and leaf](#)
 - [A.1.2. Protected link \(dual homing at one end\)](#)
 - [A.2. Augmentation pattern](#)

[Acknowledgments](#)

[Authors' Addresses](#)

1. Introduction

This draft examines the approach taken in RFC8345 to modeling the link. The draft identifies why the current unidirectional approach is not sufficient and proposes backward compatible enhancements to allow appropriate support for multipoint and bidirectional cases.

The draft is broken into several key sections:

- *Analysis of existing RFC8345: Highlighting the current unidirectional point-to-point approach taken and providing various practical examples of multipoint and highlighting experience in other standard activities and deployments.
- *Enhancement approach: Provides general details of the approach to enhancement of RFC8345
- *Basic enhancement: Provides details of YANG enhancements
- *More sophisticated enhancement: Offers an alternative YANG enhancement that further improves RFC8345
- *Further considerations: Briefly examines other areas in RFC8345 where improvements could be made
- *Conclusion: Points the way forward
- *Implementation status: Points to some implementation experience in this area
- *Appendix A.1: Provides some specific examples of multipoint links that have been encountered and continue to be encountered

*Appendix A.2: Provides a summary pattern of the enhancement using augmentation

The proposal in this draft could be used to advantage in the digital map [Digital Map] work.

2. Key terms

uni/bi: > That a single model structure supports both unidirectional and bidirectional forms where the directionality is stated by a simple enumeration.

multipoint: > That the model structure has a list of points as opposed to specifically identified individual points.

point-to-point: > That the model structure has only two points where each point has a particular specific role

3. Analysis of existing RFC8345

This section examines the approach to modeling links in RFC8345. Several snippets extracted from RFC8345 are examined. The text snippets are bounded by `[[RFCxxxx TEXT EXTRACT BEGINS]][[RFCxxxx TEXT EXTRACT ENDS]]` and the YANG snippets by `[[RFC8345 CODE EXTRACT BEGINS]][[RFC8345 CODE EXTRACT ENDS]]`.

3.1. RFC8345 section 4.2 Base Network Topology Data Model

The following text appears in the current RFC.

`[[RFC8345 TEXT EXTRACT BEGINS]]`

A link is identified by a link-id that uniquely identifies the link within a given topology. Links are point-to-point and unidirectional. Accordingly, a link contains a source and a destination. Both source and destination reference a corresponding node, as well as a termination point on that node. Similar to a node, a link can map onto one or more links (which are terminated by the corresponding underlay termination points) in an underlay topology. This is captured in the list "supporting-link".

`[[RFC8345 TEXT EXTRACT ENDS]]`

The existing RFC8345 makes a number of key points that are extracted and quoted in bullets below. Each point is followed by an observation that leads to the proposal.

*"Links are point-to-point and unidirectional."

-Observation: This restriction is the primary focus of this draft. It is proposed here that the breadth of applications can benefit significantly from a multipoint uni/bi definition as described in this draft.

*".. a link contains a source and a destination."

-Observation: But it is clear that, as the properties defined in the current RFC are "require-instance false", a link can be described in valid YANG that has no source and no destination, i.e., no termination points.

*"Both source and destination reference a corresponding node, as well as a termination point on that node."

-Observation: But in the YANG the reference can have a node alone. Under some circumstances, this may be a valid choice, although it is not clear whether the specific usages currently defined can tolerate this.

3.2. RFC8345 section 4.4.5 Cardinality and Directionality of Links

The following text appears in the current RFC.

[[RFC8345 TEXT EXTRACT BEGINS]]

The topology data model includes links that are point-to-point and unidirectional. It does not directly support multipoint and bidirectional links. Although this may appear as a limitation, the decision to do so keeps the data model simple and generic, and it allows it to be very easily subjected to applications that make use of graph algorithms. Bidirectional connections can be represented through pairs of unidirectional links. Multipoint networks can be represented through pseudonodes (similar to IS-IS, for example). By introducing hierarchies of nodes with nodes at one level mapping onto a set of other nodes at another level and by introducing new links for nodes at that level, topologies with connections representing non-point-to-point communication patterns can be represented.

[[RFC8345 TEXT EXTRACT ENDS]]

The existing RFC8345 text above provides an argument for the unidirectional solution and also provides a "work round" for more complex cases.

In the existing RFC8345 text, above, there are a number of key points that are extracted and quoted in bullets below. Each point is followed by an observation that leads to the proposal.

*"[the model] does not directly support multipoint and bidirectional links. Although this may appear as a limitation, the decision to do so keeps the data model simple and generic"

-Observation: But, as will become apparent, the multipoint uni/bi model is equally generic and arguably as simple whilst covering all cases of link including unidirectional point to point. Hence, this draft considers the restriction of the current RFC, to allow only point to point unidirectional, a limitation, not an efficiency.

-Observation: Multipoint uni/bi supports point to point (only two points declared) unidirectional (directionality selection) and hence can cover all cases covered by RFC8345 today as well as multipoint cases.

-Observation: Existing models could be transformed to the multipoint uni/bi form and, where appropriate, the multipoint uni/bi form could be used to represent multipoint cases as a set of point to point links (as is done using the current model).

*"it allows it to be very easily subjected to applications that make use of graph algorithms"

-Observation: The model is targeted at interface transfer. From practical experience it is clearly always necessary to perform some level of graph transformation prior to use by an algorithm in an application. The transformation from multipoint uni/bi has been shown to be straight forward in real solutions.

-Observation: Other transformations that are required include the interface entity becoming one or more transitional links in the topology (so as to produce a flat topology for multilayer routing). These transformation, whilst still quite straight forward are clearly more complex than the multipoint uni/bi transformation.

-Observation: Some graph applications work with bidirectional multipoint models. The multipoint construct, the hyperedge,

appears in hypergraph theory which actually better reflect the reality of networking.

*"Bidirectional connections can be represented through pairs of unidirectional links."

-Observation: Whilst true, this doubles the number of instances for many applications, which is especially significant considering that bidirectional usages are very common.

*"Multipoint networks can be represented through pseudonodes (similar to IS-IS, for example)."

-Observation: But multipoint link constructs appear from multipoint servers. There are many practical/real cases of multipoint links. Many years of deployment of management/control solutions have exposed both the reality and the benefit of multipoint constructs which is why TM Forum developed the concept and adopted it in interface models and why ONF took the lessons and adopted the same construct pattern both in the core model [ONF TR-512] and in TAPI [ONF TAPI].

-Observation: Adding a pseudo node further increases the number of instances and does not address the challenge of the partial connectability of many constructs such as root-leaf. The multipoint solution is essentially the pseudo node without the need for the links. The connectability restrictions need to be described in associated metadata (specification such as described in [ONF TR-512.7] (in the sections on ForwardingConstruct specification).

-Observation: The relative efficiency of the multipoint uni/bi solution will become clear in later sections.

3.3. From the code: "list link {"

The description for link reiterates the source destination definition and notes that the link does not model "multipoint".

[[RFC8345 CODE EXTRACT BEGINS]]

```

augment "/nw:networks/nw:network" {
  description
    "Add links to the network data model.";
  list link {
    key "link-id";
    description
      "A network link connects a local (source) node and
      a remote (destination) node via a set of the respective
      node's termination points. It is possible to have several
      links between the same source and destination nodes.
      Likewise, a link could potentially be re-homed between
      termination points. Therefore, in order to ensure that we
      would always know to distinguish between links, every link
      is identified by a dedicated link identifier. Note that a
      link models a point-to-point link, not a multipoint link.";
    leaf link-id {
      type link-id;
      description
        "The identifier of a link in the topology.
        A link is specific to a topology to which it belongs.";
    }
  }
}

```

[[RFC8345 CODE EXTRACT ENDS]]

3.4. From the code: "container source {"

The code continues with a definition of the source. This definition allows, via "require-instance false;" for either source-node or source-tp or both to be absent. Clearly, considering the definition 'path "../..../nw:node[nw:node-id=current()/../"+"source-node]/termination-point/tp-id";' the source-tp cannot be present without the source-node, but the source-node can be present without the source-tp. This may be useful in some applications, although it is not clear whether particular applications were considered (and if a link end only resolved to a node were not intentional, then it is not clear why a simple TP path was not all that was provided (as that would locate the node as well as the TP)). It is also not clear whether the opportunity to not report a source end was intended to cover a single ended link case where the destination is stated alone as the source is unknown/unresolvable. This approach has been used in other solutions where a string carried the information about the far end (as the point was outside the scope of the controller and hence the identifier was "foreign"). No description of this was found in RFC8345 and a string form of the end has not been provided.

The code provides a description for source which is ambiguous and would benefit from some improvement "Source node identifier. Must be in the same topology.". It is assumed that this "same" means must be

the "same topology as the link that it is the source of". This could be stated more clearly.

[[RFC8345 CODE EXTRACT BEGINS]]

```
container source {
  description
    "This container holds the logical source of a particular
    link.";
  leaf source-node {
    type leafref {
      path "../../nw:node/nw:node-id";
      require-instance false;
    }
    description
      "Source node identifier. Must be in the same topology.";
  }
  leaf source-tp {
    type leafref {
      path "../../nw:node[nw:node-id=current()/../../"+
        "source-node]/termination-point/tp-id";
      require-instance false;
    }
    description
      "This termination point is located within the source node
      and terminates the link.";
  }
}
```

[[RFC8345 CODE EXTRACT ENDS]]

When both source-node and source-tp are not present the structure "container source {" can be omitted from the instance as it is not a "presence container" as described in RFC6020.

[[RFC6020 TEXT EXTRACT BEGINS]]

7.5.1. Containers with Presence

YANG supports two styles of containers, those that exist only for organizing the hierarchy of data nodes, and those whose presence in the configuration has an explicit meaning.

In the first style, the container has no meaning of its own, existing only to contain child nodes. This is the default style.

For example, the set of scrambling options for Synchronous Optical Network (SONET) interfaces may be placed inside a "scrambling" container to enhance the organization of the configuration hierarchy, and to keep these nodes together. The "scrambling" node

itself has no meaning, so removing the node when it becomes empty relieves the user from performing this task.

[[RFC6020 TEXT EXTRACT ENDS]]

3.5. From the code: "container destination {"

The code continues with a definition of the destination. This is of the same form as the source and can also be absent such that a link with no ends is valid in YANG. It is not clear what case this would apply to (perhaps some transitional state).

The destination definition differs from the source in that the description for the node indicates that it must be in the "same network" and not "same topology" as in the source. One is clearly in error (minor).

[[RFC8345 CODE EXTRACT BEGINS]]

```
container destination {
  description
    "This container holds the logical destination of a
    particular link.";
  leaf dest-node {
    type leafref {
      path "../../nw:node/nw:node-id";
      require-instance false;
    }
    description
      "Destination node identifier. Must be in the same
      network.";
  }
  leaf dest-tp {
    type leafref {
      path "../../nw:node[nw:node-id=current()../../"+
        "dest-node]/termination-point/tp-id";
      require-instance false;
    }
    description
      "This termination point is located within the
      destination node and terminates the link.";
  }
}
```

[[RFC8345 CODE EXTRACT ENDS]]

4. Enhancement approach

4.1. Observations

As noted, a link can have no source and no destination. This leads to an opportunity for a simple approach to enhancement by providing an additional optional link end definition in the structure. The addition appears to be YANG backward compatible as existing point-to-point unidirectional solutions can continue to operate unchanged.

4.2. Essential Enhancement and usage

The additional optional link end definition is effectively a list of point references where the point reference definition is the same as the point reference definition in the existing source and destination structures. Each list member has, in addition to the point reference, a point-direction and a point-role.

The link also gains a link-type property that essentially references a definition of the effective internal structure of the link. The point-role is meaningful with respect to this link-type. For example, a ROOT_AND_LEAF link-type would have point-roles ROOT and LEAF and the definition of ROOT_AND_LEAF would express that information can flow from any LEAF to the ROOT and from the ROOT to any LEAF, but that flow from LEAF to LEAF is not allowed. Ideally the link-type would reference a machine interpretable specification of capability that would express these capabilities and limitations.

The point-direction property in each end definition would express the flow with respect to the link boundary so EGRESS_FROM_LINK is flowing out of the link and INGRESS_TO_LINK is flowing into the link. A BIDIRECTIONAL point supports both ingress and egress flows.

A point could be augmented with properties where appropriate for a particular application. Where the point is BIDIRECTIONAL there could be unidirectional augments, one for ingress and one for egress. A link-direction property is also added to simplify interpretation of some common cases. This property takes the values BIDIRECTIONAL (where all points are BIDIRECTIONAL), UNIDIRECTIONAL (where each point is either INGRESS_TO_LINK or EGRESS_FROM_LINK) and MIXED_DIRECTION (where there is no restriction on the mix of point-direction such that some points may be BIDIRECTIONAL and others INGRESS_TO_LINK etc.).

The machine interpretable specification is not fundamentally necessary as a traditional approach of coding an understanding of each standard link-type would work reasonably well. However, a machine interpretable specification would enable a client to deal with link-types that it had not encountered but through interpretation of the specification could "understand". The

specification could take a, somewhat verbose, form of connectivity matrix or could take the, more sophisticated and compact, form of rule system to describe interior arrangement and the effect of the link. An example rule system can be found in [ONF TR-512.7]. A fully generalized solution would need to take advantage of concepts set out in [mobo].

5. Comparison of the existing point-to-point solution with this multipoint proposal

As noted earlier, the current version of RFC8345 proposes the use of a pseudonode to deal with multipoint cases. This adds complexity and does not convey any flow restrictions without the addition of the equivalent of the specification discussed above. Clearly, if the pseudo-node is enriched to include a specification it needs to then have explicit points with roles etc. and then becomes of the form of the multipoint link discussed here, but it also requires a mass of point-to-point unidirectional links to connect it.

The multipoint uni/bi solution degenerates to point to point unidirectional where the list has only two points with one point as INGRESS_TO_LINK and the other as EGRESS_FROM_LINK. The link-type would indicate that the link is point to point unidirectional and the link-direction would be UNIDIRECTIONAL.

On this basis the multipoint solution proposed here covers all scenarios in an efficient and uniform fashion and hence the recommendation.?

6. Basic enhancement

The existing YANG solution is enhanced to include a point-list, link-type and link-direction. The YANG below also includes the correction to the source description (to say "same network"):

```
[[CODE BEGINS]]
```

```

augment "/nw:networks/nw:network" {
  description
    "Add links to the network data model.";
  list link {
    key "link-id";
    description
      "A network link connects a local (source) node and
      a remote (destination) node via a set of the respective
      node's termination points. It is possible to have several
      links between the same source and destination nodes.
      Likewise, a link could potentially be re-homed between
      termination points. Therefore, in order to ensure that we
      would always know to distinguish between links, every link
      is identified by a dedicated link identifier. Note that a
      link may model a point-to-point link or a multipoint
      link.";
    leaf link-id {
      type link-id;
      description
        "The identifier of a link in the topology.
        A link is specific to a topology to which it belongs.";
    }
    container source {
      description
        "This container holds the logical source of a particular
        link.";
      leaf source-node {
        type leafref {
          path "../.../nw:node/nw:node-id";
          require-instance false;
        }
        description
          "Source node identifier. Must be in the same network.";
      }
      leaf source-tp {
        type leafref {
          path "../.../nw:node[nw:node-id=current()]/../"+
            "source-node]/termination-point/tp-id";
          require-instance false;
        }
        description
          "This termination point is located within the source node
          and terminates the link.";
      }
    }
  }
  container destination {
    description
      "This container holds the logical destination of a
      particular link.";
  }
}

```

```

leaf dest-node {
  type leafref {
    path "../../../../../nw:node/nw:node-id";
    require-instance false;
  }
  description
    "Destination node identifier. Must be in the same
    network.";
}
leaf dest-tp {
  type leafref {
    path "../../../../../nw:node[nw:node-id=current()/../../"+
      "dest-node]/termination-point/tp-id";
    require-instance false;
  }
  description
    "This termination point is located within the
    destination node and terminates the link.";
}
}
container point-list {
  description
    "This container holds the points of a particular link.";
  list points
    key "point-id";
    leaf point-id {
      type string;
      description
        "Identifier of point in link.";
    }
  }
  leaf linked-node {
    type leafref {
      path "../../../../../nw:node/nw:node-id";
      require-instance false;
    }
    description
      "node identifier. Must be in the same network as the
      link.";
  }
  leaf linked-tp { // note that still need to deal with uni/bi
    type leafref {
      path "../../../../../nw:node[nw:node-id=current()/../../"+
        "linked-node]/termination-point/tp-id";
      require-instance false;
    }
    description
      "This termination point is located within the
      node and terminates the link.";
  }
}

```

```

leaf point-role {
  type role-of-point;
  require-instance false;
  description
    "The role of the point in the link defined in the
    link-type spec.";
}
leaf point-name {
  type string;
  require-instance false;
  description
    "The name of the point in the link";
}
leaf point-direction {
  type direction-of-point;
  require-instance false;

  description
    "The direction of the point in the link";
}
}
leaf link-type
  type type-of-link;
  require-instance false;
  description
    "The reference to the specification for the internal
    structure of the link.";
}
leaf link-direction;
  type direction-of-link;
  require-instance false;
  description
    "The collective direction of the points of the link.";
}
list supporting-link {
  key "network-ref link-ref";
  description
    "Identifies the link or links on which this link depends.";
  leaf network-ref {
    type leafref {
      path "../..../nw:supporting-network/nw:network-ref";
      require-instance false;
    }
  }
  description
    "This leaf identifies in which underlay topology
    the supporting link is present.";
}
leaf link-ref {
  type leafref {

```

```
    path "/nw:networks/nw:network[nw:network-id=current()/" +
      "../network-ref]/link/link-id";
    require-instance false;
  }
  description
    "This leaf identifies a link that is a part
    of this link's underlay. Reference loops in which
    a link identifies itself as its underlay, either
    directly or transitively, are not allowed.";
}
}
}
```


[[CODE ENDS]]

In addition to the main body of YANG, there are several new types. The enumerations should be extensible and therefore need to be modelled as identities. The base model will have general identities which may then be augmented for use in specific cases. The definition sketches below highlight the base model and, where applicable, some possible augmentations:

*role-of-point is an identity which includes SYMMETRIC, SOURCE, DESTINATION in the base model. This may be extended with ROOT, LEAF, PROTECTED, PROTECTING etc. It is likely that the property will need to be a list as there are potentially several roles and even a "named list" where the role is declared as protection-role etc.

*direction-of-point is an identity that includes INGRESS_TO_LINK, EGRESS_FROM_LINK, BIDIRECTIONAL in the base model.

*type-of-link is an identity that includes SYMMETRIC, POINT_TO_POINT in the base model. This may be extended with ROOT_AND_LEAF, DUAL_HOMED_PROTECTION etc.

*direction-of-link is an identity that includes UNIDIRECTIONAL, BIDIRECTIONAL and MIXED_DIRECTION in the base model.

7. More sophisticated enhancement

Whilst the basic enhancement appears simple and sufficient, a more sophisticated approach that improves the integrity of the existing model is also proposed here.

In this approach the existing source and destination structures are extracted and then augmented back into the link. As the augment is in the same module there is no namespace change. Whilst not simply YANG backward compatible, this will produce the same instance structures (in JSON etc.) and will support any augmentation of source and destination in any current usage.

The benefit of this adjustment is that the inclusion of the points can be controlled based upon feature support which better separates the structures that support the existing capability from those that support the new capability.

The new point-list is also added in via augmentation but with a different feature control. The existing YANG solution is enhanced to include a point-list, link-type and link-direction. The YANG below also includes the correction to the source description (to say "same network"):

[[CODE BEGINS]]

```
augment "nw:networks/nw:network/nw:link" {
  description
    "Add source to link where the link is traditional
    uni-point-to-point";
  when 'derived-from-or-self(some useful property,
    "UNI_POINT_TO_POINT)';
  if-feature uni-point-to-point;
  container source {
    uses source-properties;
    description "none";
  }
}
```

```
augment "nw:networks/nw:network/nw:link" {
  description
    "Add destination to link where the link is traditional
    uni-point-to-point";
  when 'derived-from-or-self(some useful property,
    "UNI_POINT_TO_POINT)';
  if-feature uni-point-to-point;
  container destination {
    uses destination-properties;
    description "none";
  }
}
```

```
augment "nw:networks/nw:network/nw:link" {
  description
    "Add point-list for uniform support of point-to-point and
    multipoint";
  when 'derived-from-or-self(some useful property, "UNI_BI_MULTI)';
  if-feature uni-bi-multi;
  container point-list {
    uses point-list-properties;
    description "none";
  }
}
```

```
augment "nw:networks/nw:network/nw:link" {
  description
    "Add multipoint properties for uniform support of
    point-to-point and multipoint";
  when 'derived-from-or-self(some useful property, "UNI_BI_MULTI)';
  if-feature uni-bi-multi;
  container multipoint-link-properties {
    uses multipoint-link-properties;
    description "none";
  }
}
```

```
augment "/nw:networks/nw:network" {
  description
```

```

    "Add links to the network data model.";
list link {
  key "link-id";
  description
    "A network link connects a local (source) node and
    a remote (destination) node via a set of the respective
    node's termination points. It is possible to have several
    links between the same source and destination nodes.
    Likewise, a link could potentially be re-homed between
    termination points. Therefore, in order to ensure that we
    would always know to distinguish between links, every link
    is identified by a dedicated link identifier. Note that a
    link may model a point-to-point link or a multipoint
    link.";
  leaf link-id {
    type link-id;
    description
      "The identifier of a link in the topology.
      A link is specific to a topology to which it belongs.";
  }

  list supporting-link {
    key "network-ref link-ref";
    description
      "Identifies the link or links on which this link depends.";
    leaf network-ref {
      type leafref {
        path "../..../nw:supporting-network/nw:network-ref";
        require-instance false;
      }
      description
        "This leaf identifies in which underlay topology
        the supporting link is present.";
    }
    leaf link-ref {
      type leafref {
        path "/nw:networks/nw:network[nw:network-id=current()]/"+
          "../network-ref]/link/link-id";
        require-instance false;
      }
      description
        "This leaf identifies a link that is a part
        of this link's underlay. Reference loops in which
        a link identifies itself as its underlay, either
        directly or transitively, are not allowed.";
    }
  }
}
}
}
}

```

```

grouping source-properties {
  description
    "This grouping holds the logical source of a particular
    link.";
  leaf source-node {
    type leafref {
      path "../../nw:node/nw:node-id";
      require-instance false;
    }
    description
      "Source node identifier. Must be in the same network.";
  }
  leaf source-tp {
    type leafref {
      path "../../nw:node[nw:node-id=current()../../"+
        "source-node]/termination-point/tp-id";
      require-instance false;
    }
    description
      "This termination point is located within the source node
      and terminates the link.";
  }
}

```

```

grouping destination-properties {
  description
    "This container holds the logical destination of a
    particular link.";
  leaf dest-node {
    type leafref {
      path "../../nw:node/nw:node-id";
      require-instance false;
    }
    description
      "Destination node identifier. Must be in the same
      network.";
  }
  leaf dest-tp {
    type leafref {
      path "../../nw:node[nw:node-id=current()../../"+
        "dest-node]/termination-point/tp-id";
      require-instance false;
    }
    description
      "This termination point is located within the
      destination node and terminates the link.";
  }
}

```

```

}

grouping point-list-properties {
  description
    "This container holds the points of a particular link.";
  list points
    key "point-id";
    leaf point-id {
      type string;
      description
        "Identifier of point in link.";
    }
    leaf linked-node {
      type leafref {
        path "../.../nw:node/nw:node-id";
        require-instance false;
      }
      description
        "The node identifier. Must be in the same network.";
    }
    leaf linked-tp { // note that still need to deal with uni/bi
      type leafref {
        path "../.../nw:node[nw:node-id=current()/../"+
          "linked-node]/termination-point/tp-id";
        require-instance false;
      }
      description
        "This termination point is located within the
        node and terminates the link.";
    }
    leaf point-role {
      type role-of-point;
      description
        "The role of the point in the link";
    }
    leaf point-name {
      type string;
      description
        "The name of the point in the link";
    }
    leaf point-direction {
      type direction-of-point
      description
        "The direction of the point in in the link";
    }
  }
}

grouping multipoint-link-properties {
  description

```

```
    "This container holds the properties of a multipoint link.";
leaf link-type
  type type-of-link;
  require-instance false;
  description
    "The reference to the specification for internal
      structure of the link";
}
leaf link-direction;
  type direction-of-link;
  require-instance false;
  description
    "The collective direction of the points of the link";
}
}
```

[[CODE ENDS]]

8. Further considerations

This section points to other related areas of consideration where each one could either be covered by this draft as it evolves or could be the basis for new drafts.

8.1. Termination direction

The termination point could benefit from a direction statement as some terminations are inherently unidirectional and others bidirectional. Termination direction is a capability statement. Termination state can be different for the ingress/receive direction from the egress/transmit direction. Termination direction is challenging in that a termination has both an upper and a lower side (orientation as per overlay and underlay). Both sides may have both ingress and egress. Work has been done in several external bodies (e.g., ONF in [ONF TR-512]) on this challenge and that input should be sought prior to embarking on this addition.

8.2. Specification of capability

A termination point represents the presence of a capability to deal with flows. The termination point is silent on the actual capability. The capability of a termination point is dependent upon the underlying functionality supporting it. Functionality tends to be systematically deployed such that each termination point is of a type where there are many instances of that type in a deployment and where each termination point of a type has the same characteristic to each other. For any particular type of termination, its capability is invariant in specific value for some properties, invariant in range of values for some properties and invariant in algorithm to determine value for some properties. The statement of invariants per type is best stated in a single place and referenced by each instance. This statement could be called a type specification (as in [ONF TR-512]). Clear statement of range etc. benefits from work pointed to in [mobo]. It is suggested here that this aspect is vital for other work such as that in the area of digital twin such that it would potentially become part of the [Digital Map].

8.3. Links between networks/subnetworks/domains

The current definition in RFC8345 is limited to links within a network. There are many cases where links pass between networks. A challenge, tackled by other bodies in similar work, is the representation of a link, within a controller, that passes from a network that is in the view of that controller to a network that is not (or between two parts of what could be considered as one network

by an external observer, where only one part is in the view of the controller). Work should be carried out to develop inter-network link modeling where that modeling accounts for both the case where all ends of the link are in the view of a single controller and the case where some ends are not in the view of a controller that is having to provide the representation.

Note that a "foreign" identifier in one or more ends may be the appropriate solution as touched on earlier in this draft.

8.4. Richness of navigation

Not all possible navigations through the topology are defined in the model. There is always a dilemma here. Should the model show all possible navigations such that every relationship becomes two way (e.g., termination point relates to link end and link end to termination point), as of course, that may be what is required in an application that navigates the topology, or should the model provide only one direction and leave it to the application to populate the other (obvious) reverse relationship if it needs it. When considering transfer of information on an interface, it is redundant to send both directions of navigation and, if the entities which now refer to each other are propagated non-transactionally, there can be a temporary inconsistency (which, of course, is where eventual consistency comes in). There is a larger loop version of this challenge where there is a minimal set of relationships that sufficiently describes the topology, but where that may differ from the specific incomplete set (necessary for a particular application). As noted earlier, it is "always" the case that an application needs to do some transformation on the representation of semantics received from some other control component. It is important to agree where the intention is to provide a minimal set of relationships from which all others can be derived, a full set of all possible relationships which can then be pruned, or something in between (which is where we normally end up when the music stops).

8.5. Clarification of relationship roles

In this draft, it is recognised that there is a role of a point in the link (e.g., root in a root/leaf configuration). Other relationships may also require a similar role clarification. In work in other bodies (e.g., [ONF TR-512]), it was recognised that the fully functional representation of the termination/interface/etc. had potentially complex relationships to other equivalents and to links/connections. This leads to a consideration that all representations of functional entities could benefit from a modeling treatment using the component/system pattern [ONF TR-512.A.2]. This area has not been fully explored and at this stage appears to add significant, potentially opaque, complexity to many model areas. It

should be noted however, that this is the underpinning of the "points & roles" model for link proposed here that is clearly highly beneficial and very transparent.

8.6. More than just links and termination points

In work in other bodies in this area, it has been recognised that there is a general model of potential for flow consistent with that set out in RFC8345, but that there is also a general model of termination function and flow. Work on this area to improve coherence in modeling would be highly beneficial for the [Digital Map].

8.7. Layering and sub-layering

Other bodies have grappled with the challenge of defining what a layer really is. This area requires further consideration to ensure it is clear in RFC8345. As this is clarified, it may become apparent that better indication of layering is required on the specific entities of RFC8345.

9. Conclusion

This draft has provided a brief analysis of the current unidirectional point-to-point approach to modeling of the link in RFC8345, has highlighted why this is not sufficient and has made a proposal to enhance RFC8345 YANG to support multipoint uni/bi links where two alternative enhancement approaches were offered, both of which are backward compatible.

It is recommended that the enhancement be made, however, whether to use the simple or more sophisticated approach requires further assessment. This assessment should be carried out without delay as the enhancement could significantly benefit the digital map [Digital Map] work as well as other modeling activities.

The points highlighted under "Further considerations" should also be assessed for value and urgency, and work should be commenced to define any necessary adjustments.

10. Implementation Status

This section has been included to emphasize implementation experience in this area. There are currently no implementations of the specific proposal detailed here, but there are many implementations that take advantage of multipoint uni/bi connectivity and topology models.

10.1. Standards driven implementations

Multipoint uni/bi appears in several TMF standards such as MTNM (Multi-Technology Network Management) [TMF MTNM], which defines interfaces for interaction between a network managers and sub-network/element managers, where several entities including TopologicalLink and SubNetworkConnection use a multipoint uni/bi representation. These models date back to the early 2000s and the standards were deployed by many vendors.

More recently ONF adopted the model in core work [ONF TR-512] and TAPI [ONF TAPI] and there are implementations available that take advantage of both multipoint uni/bi connections and multipoint uni/bi links. Some implementations have been approved by TIP MUST [TIP MUST]. Major implementations of TAPU are proprietary at this point, but interoperability evaluations are ongoing between products from various vendor using the TAPI standards initially hosted by OIF (as proof-of-concept exercises) [OIF PoC] but more recently as part of operator deployment. Many of these products also take advantage of multipoint uni/bi models internally.

10.2. Proving the model

It is anticipated that a PoC activity to exercise the proposal be carried out as part of the digital map work [Digital Map].

11. Security Considerations

None

12. IANA Considerations

This document has no IANA actions.

13. Informative References

[ONF TR-512] Open Networking Foundation TR-512 Core Information Model (CoreModel) v1.5 - https://opennetworking.org/wp-content/uploads/2021/11/TR-512_v1.5_OnfCoreIm-info.zip (also published by ITU-T as G.7711 at <https://www.itu.int/rec/T-REC-G.7711/recommendation.asp?lang=en&parent=T-REC-G.7711-202202-I>)

[ONF TR-512.7] (in [ONF TR-512] Model Description Document) TR-512.7 Specification

[ONF TR-512.8] (in [ONF TR-512] Model Description Document) TR-512.8 Control

[ONF TR-512.A.2] (in [ONF TR-512] Model Description Document) TR-512.A.2 Appendix: Model Structure, Patterns and Architecture

[ONF TAPI] Open Networking Foundation Transport API SDK 2.4.0 - <https://github.com/OpenNetworkingFoundation/TAPI/releases/tag/v2.4.0>

[TMF MTNM] TeleManagement Forum MultiTechnology Network Management - <https://www.tmforum.org/resources/collection/mtnm-4-5/>

[TIP MUST] Telecoms Infra Project Mandatory Use Cases for SDN Transport - https://cdn.brandfolder.io/D8DI15S7/at/kzt845vb2q9r2twr8jtgqm4/TIP_00PT_MUST_Use-Cases-and-Technical-Requirements-for-Wireless-Backhaul-SDN-Domain-Controller--Network-Equipment-FINAL-GREEN-ACCESSv20.pdf

[OIF PoC] Optical Interworking Forum Proof of Concepts - <https://www.oiforum.com/technical-work/2018-sdn-transport-api-interopability-demo/>

[Digital Map] Digital Map - <https://www.ietf.org/id/draft-havel-opsawg-digital-map-00.txt>

[mobo] Modelling Boundaries - <https://datatracker.ietf.org/doc/draft-davis-netmod-modelling-boundaries/>

Appendix A. Appendix A

Note also that in some multipoint contexts there is a link scale issue and tp referencing the link it belongs to may be a better orientation. It would then need the role in the link etc.

A.1. Examples of multipoint

There are many examples of multipoint links

A.1.1. Root and leaf

The following diagram shows a sketch of a root and leaf link where the bidirectional points of the connection are represented by the pair of symbols ">" and "<" which indicate the ingress and egress flows of the bidirectional point.

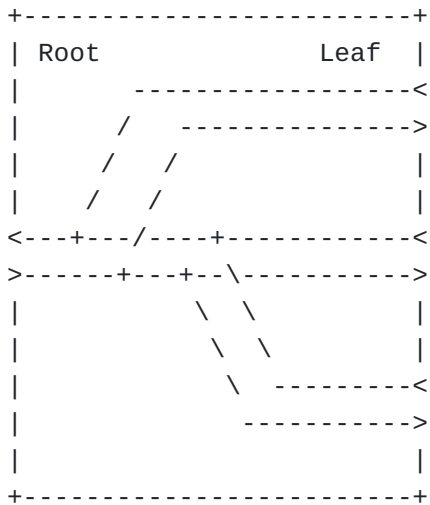


Figure 1: A Root and Leaf Link

Flow from root to leaf and from leaf to root is allowed. Flow from leaf to leaf is not allowed. These restrictions can be stated in a root-leaf specification structure that defines the allowed flows in terms of rules where the point role (root or leaf) is used to identify applicable rules etc.

A.1.2. Protected link (dual homing at one end)

The following diagram shows a sketch of a protected link where the bidirectional points of the connection are represented by the pair of symbols ">" and "<" which indicate the ingress and egress flows of the bidirectional point. The switch is shown as "x" at the common point and "o" at the two selectable points. The switch is currently set to take signal from the main protecting point.

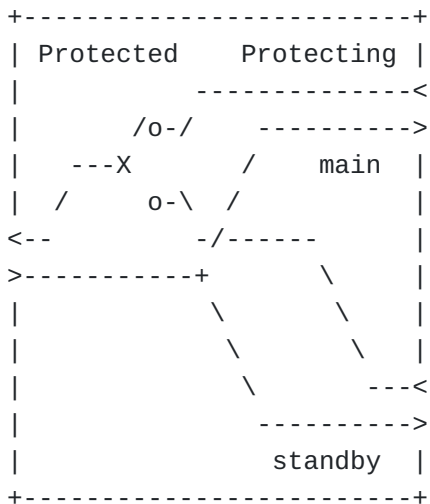


Figure 2: A Protected Link with exposed dual homing

Flow from protected to protecting and from protecting to protected is allowed. Flow from protecting to protecting is not allowed. Depending upon the detailed type, it is possible for the protected port to feed signal to both protecting ports. The protected port is fed from either main or standby protecting port depending upon signal quality.

A.2. Augmentation pattern

RFC 8345 pruned/adjusted snippet.

[[CODE BEGINS]]

```
augment "/nw:networks/nw:network" {
  list link {
    key "link-id";
    leaf link-id {
      type link-id;
    }
    container source {
      leaf source-node {
        type string;
      }
    }
  }
}
```

[[CODE ENDS]]

Alternative form:

[[CODE BEGINS]]

```
augment "nw:networks/nw:network/nw:link" {
  container source {
    uses source-properties;
  }
}

augment "/nw:networks/nw:network" {
  list link {
    key "link-id";
    leaf link-id {
      type link-id;
    }
  }
}

  grouping source-properties {
    leaf source-node {
      type string;
    }
  }
}
```

[[CODE ENDS]]

A JSON instance conformant to the first form above is also conformant to the second, alternative, form.

Acknowledgments

Authors' Addresses

Nigel Davis
Ciena

Email: ndavis@ciena.com

Olga Havel
Huawei

Email: olga.havel@huawei.com

Benoit Claise
Huawei

Email: benoit.claise@huawei.com