NETMOD Versioning Design Team Update

NETMOD WG

July 22, 2019

Netmod YANG Version Design Team

Presenting: Joe Clarke, Rob Wilton, Balazs Lengyel

Agenda

- **Requirements Draft update – Joe Clarke (5 mins)**
  draft-ietf-netmod-yang-versioning-reqs-01
- **Design Team update & Solution overview – Rob Wilton (20 mins)**
  draft-verdt-netmod-yang-solutions-01
- **Updated YANG Module Revision Handling – Balazs Lengyel (25 mins)**
  draft-verdt-netmod-yang-module-versioning-00
- **Next steps – Rob Wilton (5 min)**

draft-ietf-netmod-yang-versioning-reqs-01

Netmod WG

July 22, 2019

Netmod YANG Version Design Team
Joe Clarke (presenting)

Changes Since Design Team Draft

- Adopted by working group
  - Draft-verdt-netmod-yang-versioning-reqs ⫢ draft-ietf-netmod-yang-versioning-reqs
- Changed requirement 1.4 text based on feedback from IETF 104

Requirement 1.4

OLD:

“The solution MUST allow for backwards-compatible enhancements and bug fixes, as well as non-backwards- compatible bug fixes in non-latest-release modules.”

NEW:

“The solution MUST be able to express when non-backwards-compatible changes have occurred between two revisions of a given YANG module.”

Design Team update

Netmod WG

July 22, 2019

Netmod YANG Version Design Team
Rob Wilton (presenting)

Design Team Update

- Continued with semi-regular weekly meetings since IETF 104
- Continued thanks to all DT members: Balazs Lengyel, Benoit Claise, Ebben Aries, Jason Sterne, Joe Clarke, Juergen Schoenwaelder, Mahesh Jethanandani, Michael (Wangzitao), Qin Wu, Reshad Rahman, Rob Wilton
- DT output since 104:
  1. Updated solution overview draft
  2. New/updated YANG Module Revision Handling draft (main focus)

     We think that this draft accommodates all the core feedback received during the versioning related discussions and meetings at IETF 104

YANG Versioning Solution Overview

draft-verdt-netmod-yang-solutions-01

NETMOD WG

July 22, 2019

Netmod YANG Version Design Team

Presenting: Rob Wilton

YANG Versioning Solution Overview

- Temporary document to help guide reviewers through the YANG versioning solution space
  - Not intending on taking this to RFC
- Further details for some solution areas follow as separate presentations:
  (i) Updated module revisioning handling
  (ii) YANG packages (after the DT updates)
  (iii) Versioning selection (after the DT updates)

YANG Versioning Solution Overview

Proposed full complete versioning solution is made up of five drafts (at different levels of maturity):

1. Updated YANG Module Revision Handling (draft-verdt-netmod-yang-module-versioning-00)
2. Module semantic version number scheme (to be derived from draft-verdt-netmod-yang-semver-00)
3. Versioned YANG packages (draft-rwilton-netmod-yang-packages-01)
4. Protocol operations for package version selection (draft-wilton-netmod-yang-ver-selection-00)
5. YANG schema comparison tooling (deferred for now)

Updated YANG Module Revision Handling draft-verdt-netmod-yang-module-versioning-00

- Core enhancements to YANG to:
  - *Allow* non linear module development (e.g. for bugfixes)
  - *Document* when non-backwards-compatible changes have occurred in revision history
- Module revisions are still uniquely identified by revision date, but allow a freeform text label to be associated with a revision
  - *Label could be semver based or a different scheme*
- “Import by revision-date or derived” (using revision history)
- Define backwards-compatible vs non-backwards-compatible changes
- Clarify/improve YANG “status” handling
- Refresh guidelines for updating YANG modules

Semantic version number scheme
(Not written yet)

The plan:

- Define a semantic versioning scheme that allows for bugfixes to released software assets.
- Based on the algorithm defined in draft-verdt-netmod-yang-semver-00
- Does not define what constitutes BC vs NBC changes
- **Can be used as a “revision label”**
- Could also be used to version other software assets (e.g. YANG packages)

YANG packages
(draft-rwilton-netmod-yang-packages-01, but requires updates)

Aims:

- A YANG package identifies a set of YANG modules, and dependencies
- Packages can be versioned liked modules
- Available offline and via YANG library augmentation
- Could simplify conformance:
  - check packages rather than lists of modules
  - implement package versions rather than arbitrary module versions
- Separate presentation to follow

Protocol operations for package version selection (draft-wilton-netmod-yang-ver-selection-00, early draft)

Aims:

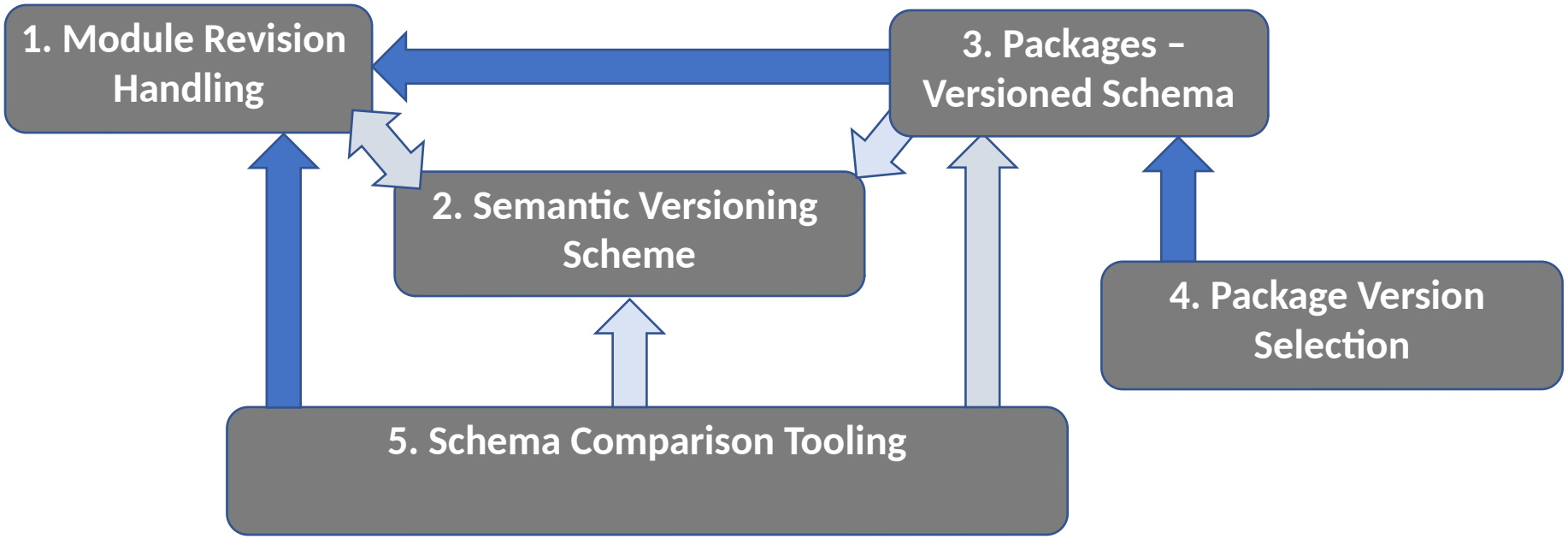
- Allow clients to select which package version(s) to use in device interactions:
  - E.g. *vendor-modules@6.5.1*, *vendor-modules@6.4.3*, *ietf-modules@1.8.0*, or *oc-modules@2.3.1*
- Server might only support one package version for entire management interface
- Or servers might be able to support different package versions on different sessions
- Separate presentation to follow

YANG schema comparison tooling
(No draft yet, lower priority)

Aims:

- Define an algorithm to compare two schema trees, detecting backwards-compatible vs non-backwards compatible changes
- Result could be given at the schema or per data-node level
- Could take into account features, deviations, and the subset of the schema being used
- Could be useful validating YANG history and semantic version numbers
- May consider annotations for YANG stmts that are not easily comparable (descriptions, xpath expressions, regular expressions)

YANG versioning solution – dependencies:



Dependency

Possible user dependency

Updated YANG Module Revision Handling

draft-verdt-netmod-yang-module-versioning-00

NETMOD WG

July 22, 2019

Netmod YANG Version Design Team

Presenting: Balazs Lengyel

Updated YANG Module Revision Handling draft-verdt-netmod-yang-module-versioning-00

- Derived from draft-verdt-netmod-yang-semver-00, but without the semver part
- We think that this draft accommodates all the core feedback received during the versioning related discussions and meetings at IETF 104
- Updates existing RFCs
- A collection of versioning related enhancements
- Versioning applies to YANG Modules only – submodules are implicitly versioned as part of the including YANG module

Updates to existing YANG RFCs

1. The document updates YANG 1.1 [RFC7950] section 11 for:
   - modifications to YANG revision handling and update rules
   - a YANG extension statement to do import by derived revision

2. The document updates YANG library [RFC8525] section 3:
   - to provides guidance on how to resolve ambiguous import revisions

3. This document updates YANG author guidelines [RFC8407] sec 4.7:
   - to provide guidelines on managing the lifecycle of YANG modules that may contain NBC changes and a branched revision history

Changes/Enhancements

1. Extension statement to indicate NBC changes in revision history
2. Allows non-linear revision history
3. Revision labels
4. Import revision-or-derived extension statement
5. Defines BC and NBC changes for YANG modules
6. Updated YANG “status” handling
7. YANG library augmentations/updates
8. YANG instance data versioning considerations
9. Updated guidelines for updating modules

“rev:nbc-changes” extension statement to indicate NBC changes in revision history

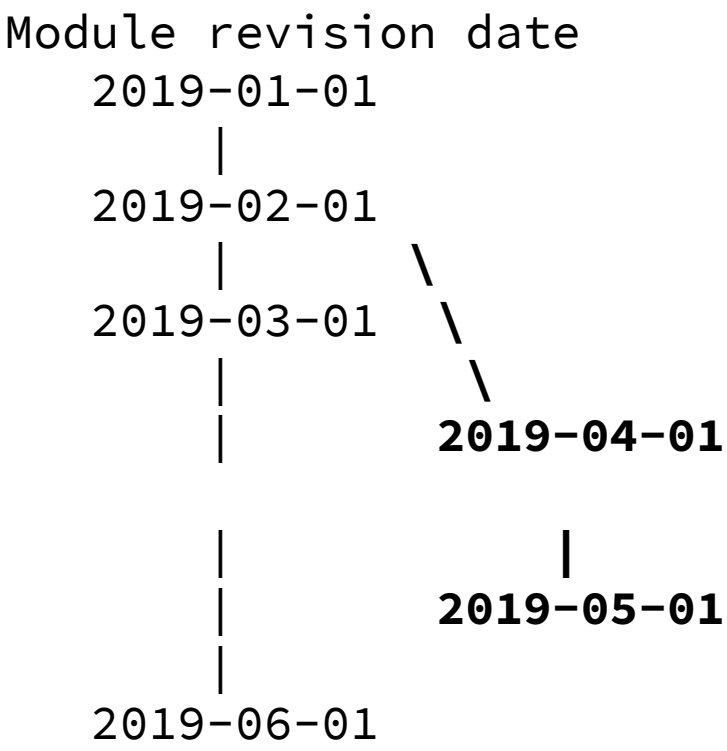
```
revision 2019-06-01 {
  rev:nbc-changes;
  description "Add new functionality.";
}

revision 2019-04-01 {
  description
    "Add new functionality. Remove some deprecated nodes.";
}

revision 2019-02-01 {
  rev:nbc-changes;
  description "Apply bugfix to pattern statement";
}

revision 2019-01-01 {
  description "Initial revision";
}
```

Only backwards compatible changes between this revision and the previous revision

Non backwards compatible changes between this revision and the previous revision

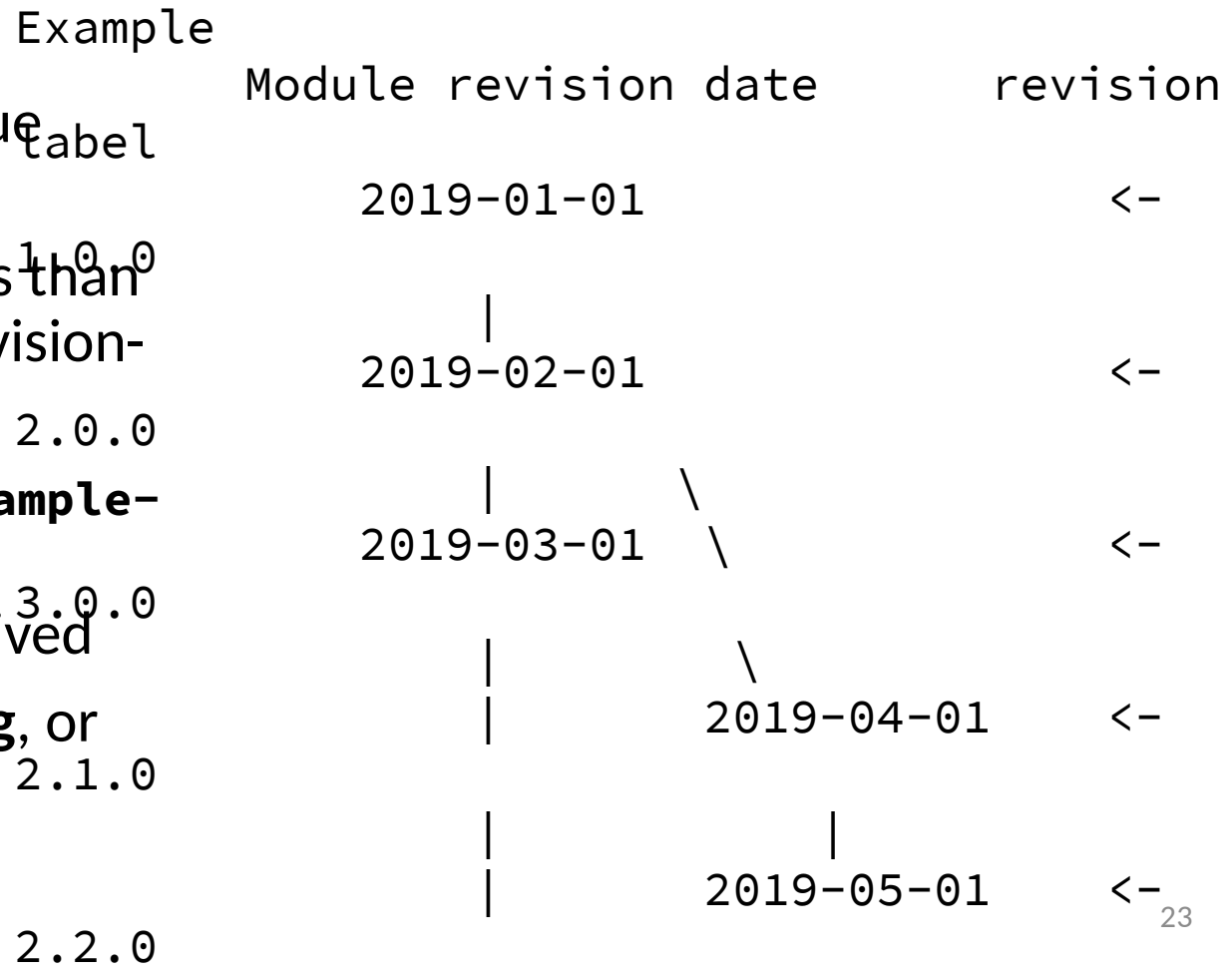
Allows non-linear revision history

- Not actually disallowed by RFC 7950!
- No limit to the amount of branching that could be expressed!
- **Does not imply arbitrary branching is good or recommended**
- Revision history represents the path from that revision to the root of the revision tree



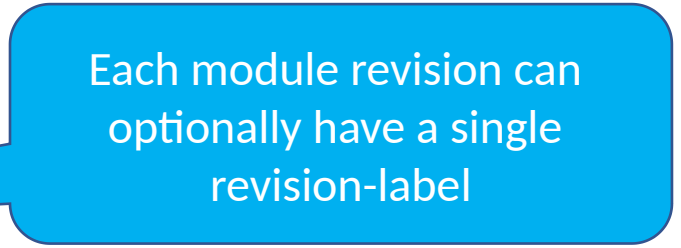
Revision labels

- Optional *alternative* text identifier for a revision
- Revision-date is still the primary unique identifier for a YANG module revision
- Minimal constraints on its format (less than 255 chars, no spaces or ‘@’s, not a revision-date)
- Can be used in module file name: **`example-foo@3.1.0.yang`**
- Can be used in import revision-or-derived
- Could by used for **semantic versioning**, or **other revision identifier**



Revision labels (2)

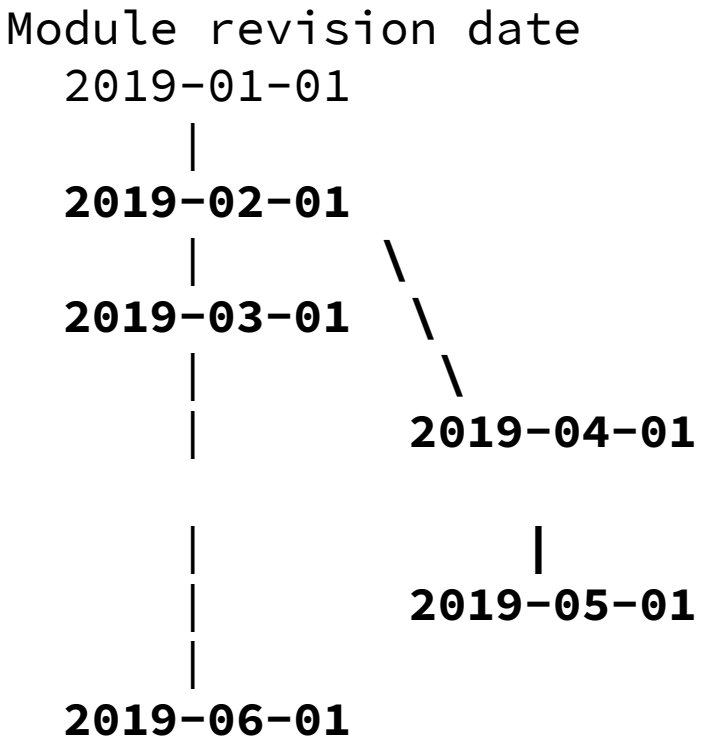
```
revision 2019-06-01 {
  rev:revision-label "3.1.0";
  description "Add new functionality.";
}

revision 2019-04-01 {
  rev:revision-label "3.0.0";
  rev:nbc-changes;
  description
    "Add new functionality. Remove some deprecated nodes.";
}

revision 2019-02-01 {
  rev:revision-label "2.0.0";
  rev:nbc-changes;
  description "Apply bugfix to pattern statement";
}

revision 2019-01-01 {
  rev:revision-label "1.0.0";
  description "Initial revision";
}
```



Import revision-or-derived extension statement

- “Import by revision” is too strict
- “Basic import” is not strict enough
- Import **ver:revision-or-derived** selects any versions that see the selected version in the history
- Can use r**evision-date or label** – the meaning is the same
- Multiple **ver:revision-or-derived** stmts allowed

```
import example-module {
  ver:revision-or-derived 2019-02-
01;
}

import example-module {
  ver:revision-or-derived 1.0.0;
}
```



Define BC and NBC changes for YANG modules

Backwards-compatible:

Mostly RFC 7950 section 11 rules except:

- Deprecating a node is BC (*assumes such nodes are still implemented*)
- Obsoleting node is NBC
- Data definition statements can be reordered if they don't change the ordering in "rpc" or "input" substatements

Non-backwards-compatible:

- Any changes that are not classified as backwards compatible ^

Updated YANG “status” handling

- Change “status deprecated” to mean that the node SHOULD be implemented, or otherwise it must be deviated
- Change “status obsolete” to mean that the node SHOULD NOT be implemented
- Two state leaves augmented to YANG library to report servers compliance.
- “rev:status-description” extension – to add a description to a status change (e.g., why it is changing, when it will be removed).

YANG library augmentations/updates

1. Two leaves added to YANG library to report server status handling:

   `deprecated-nodes-implemented`

   `obsolete-nodes-absent`

   Note - if leaves aren't present then server behaviour is undefined - as per today

2. Define which version of a imported module should be chosen, if there are multiple versions to choose from:
   1. If the module is implemented, then the implemented revision
   2. If the module is import-only, then the latest applicable module revision that complies with the import statement is used

   Note – as Martin has pointed out, this is really a bug fix to RFC 7950/8525

YANG instance data versioning considerations

- Instance data is not versioned (as part of this work)
- But instance data may relate to a schema that is versioned (e.g. via YANG library, or YANG packages)

Updated Guidelines when updating modules (for module authors)

- RECOMMENDED to minimize NBC changes
- SHOULD mitigate impact on clients if NBC changes occur
- Removing old revision statements from the revision history could break imports, and hence SHOULD be avoided
- SHOULD use “ver:revision-or-derived” to indicate revision dependencies between modules
- Submodule includes MUST use “revision-date” to include specific submodule revisions
- The draft provides guidance and examples of appropriate steps to make NBC updates.

Updated Guidelines when updating modules (considerations for clients)

- Clients SHOULD be liberal on that they receive (e.g. if the range of an operational value has increased)
- Clients SHOULD monitor changes to published YANG modules. Clients should not update to a module revision with NBC changes without understanding any potential impact of the changes
- Clients SHOULD plan to update to new nodes in a timely fashion if nodes become deprecated.

YANG Versioning DT – Next steps

NETMOD WG

July 22, 2019

Netmod YANG Version Design Team

Presenting: Rob Wilton

Next Steps

Working through the different solution parts:

1. Seek WG adoption of draft-verdt-netmod-yang-module-versioning-00
2. Initial draft of “Semantic version number scheme”, derived from draft-verdt-netmod-yang-semver-00
3. Perhaps have the DT work on refining the YANG packages and version selection drafts.