MODULE *Config*

INSTANCE *Naturals*

INSTANCE *FiniteSets*

INSTANCE *Sequences*

LOCAL INSTANCE *TLC*

An empty constant

CONSTANT *Nil*

Transaction type constants

CONSTANTS
Change,
Rollback

Transaction isolation constants

CONSTANTS
ReadCommitted,
Serializable

Phase constants

CONSTANTS
Initialize,
Validate,
Abort,
Commit,
Apply

$Phase \triangleq$
{*Initialize*,
Validate,
Abort,
Commit,
Apply}

Status constants

CONSTANTS
InProgress,
Complete,
Failed

$State \triangleq$
{*InProgress*,
Complete,

Failed}

State constants

CONSTANTS
Pending,
Validated,
Committed,
Applied,
Aborted

Status $\triangleq$
{*Pending*,
Validated,
Committed,
Applied,
Aborted}

CONSTANTS
Valid,
Invalid

CONSTANTS
Success,
Failure

The set of all nodes

CONSTANT *Node*

Target is the set of all targets and their possible paths and values.

Example:
Target $\triangleq$
[*target*1 $\mapsto$
[*persistent* $\mapsto$ FALSE, *values* $\mapsto$ [
*path*1 $\mapsto$ {"*value*1", "*value*2"},
*path*2 $\mapsto$ {"*value*2", "*value*3"}]],
*target*2 $\mapsto$
[*persistent* $\mapsto$ TRUE, *values* $\mapsto$ [
*path*2 $\mapsto$ {"*value*3", "*value*4"},
*path*3 $\mapsto$ {"*value*4", "*value*5"}]]]

CONSTANT *Target*

Configuration update/rollback requests are tracked and processed through two data types. Transactions represent the lifecycle of a single configuration change request and are stored in an append-only log. Configurations represent the desired configuration of a *gNMI* target based on the aggregate of relevant changes in the *Transaction* log.

TYPE Type ::= *type* $\in$

```
  {Change,
   Rollback}

TYPE Phase ::= phase ∈
  {Initialize,
   Validate,
   Abort,
   Commit,
   Apply}

TYPE State ::= state ∈
  {InProgress,
   Complete,
   Failed}

TYPE Status ::= status ∈
  {Pending,
   Validated,
   Committed,
   Applied,
   Aborted}

TYPE Isolation ::= isolation ∈
  {ReadCommitted,
   Serializable}

TYPE Transaction ≜
  [type      ::= type ∈ Type,
   isolation ::= isolation ∈ Isolation
   change ::=
    [target ∈ SUBSET (DOMAIN Target) ↦
      [path ∈ SUBSET (DOMAIN Target[target].values) ↦
        [value ::= value ∈ STRING,
         delete ::= delete ∈ BOOLEAN ]]],
   rollback ::= index ∈ Nat,
   targets ::= targets ∈ SUBSET (DOMAIN Target)
   phase     ::= phase ∈ Phase,
   state     ::= state ∈ State,
   status ::= status ∈ Status]

TYPE Proposal ≜
  [type      ::= type ∈ Type,
   change    ::=
    [index ::= index ∈ Nat,
     values ::=
      [path ∈ SUBSET (DOMAIN Target[target].values) ↦
        [value ::= value ∈ STRING,
         delete ::= delete ∈ BOOLEAN ]]],
   rollback ::=
    [index ::= index ∈ Nat,
     values ::=
      [path ∈ SUBSET (DOMAIN Target[target].values) ↦
        [value ::= value ∈ STRING,
```

```
        delete ::= delete ∈ BOOLEAN ]]],
    dependency ::= [index ∈ Nat],
    phase      ::= phase ∈ Phase,
    state      ::= state ∈ State]
TYPE Configuration ≜
  [config ::=
    [index ::= index ∈ Nat,
     term ::= term ∈ Nat,
     values ::=
      [path ∈ SUBSET (DOMAIN Target[target]) ↦
        [value ::= value ∈ STRING,
         index ::= index ∈ Nat,
         deleted ::= delete ∈ BOOLEAN ]]],
    proposal ::= [index ::= index ∈ Nat],
    commit ::= [index ::= index ∈ Nat],
    target ::=
     [index ::= index ∈ Nat,
      term ::= term ∈ Nat,
      values ::=
       [path ∈ SUBSET (DOMAIN Target[target]) ↦
         [value ::= value ∈ STRING,
          index ::= index ∈ Nat,
          deleted ::= delete ∈ BOOLEAN ]]],
    state ::= state ∈ State]
```

A transaction log. Transactions may either request a set of changes to a set of targets or rollback a prior change.

VARIABLE *transaction*

A record of per-target proposals

VARIABLE *proposal*

A record of per-target configurations

VARIABLE *configuration*

A record of target states

VARIABLE *target*

A record of target *masterships*

VARIABLE *mastership*

$vars \triangleq \langle transaction, proposal, configuration, mastership, target \rangle$

Transaction ≜ INSTANCE *Transaction*
Proposal ≜ INSTANCE *Proposal*
Configuration ≜ INSTANCE *Configuration*
Southbound ≜ INSTANCE *Southbound*
Northbound ≜ INSTANCE *Northbound*

Formal specification, constraints, and theorems.

$$
\begin{array}{l}
\mathit{Init} \triangleq \\
\quad \land \mathit{Transaction}\,!\,\mathit{Init} \\
\quad \land \mathit{Proposal}\,!\,\mathit{Init} \\
\quad \land \mathit{Configuration}\,!\,\mathit{Init} \\
\quad \land \mathit{Northbound}\,!\,\mathit{Init} \\
\quad \land \mathit{Southbound}\,!\,\mathit{Init}
\end{array}
$$

$$
\begin{array}{l}
\mathit{Next} \triangleq \\
\quad \lor \land \mathit{Transaction}\,!\,\mathit{Next} \\
\qquad \land \text{UNCHANGED } \langle \mathit{configuration},\ \mathit{target},\ \mathit{mastership} \rangle \\
\quad \lor \land \mathit{Proposal}\,!\,\mathit{Next} \\
\qquad \land \text{UNCHANGED } \langle \mathit{transaction} \rangle \\
\quad \lor \land \mathit{Configuration}\,!\,\mathit{Next} \\
\qquad \land \text{UNCHANGED } \langle \mathit{transaction},\ \mathit{proposal} \rangle \\
\quad \lor \land \mathit{Northbound}\,!\,\mathit{Next} \\
\qquad \land \text{UNCHANGED } \langle \mathit{proposal},\ \mathit{configuration},\ \mathit{target},\ \mathit{mastership} \rangle \\
\quad \lor \land \mathit{Southbound}\,!\,\mathit{Next} \\
\qquad \land \text{UNCHANGED } \langle \mathit{transaction},\ \mathit{proposal},\ \mathit{configuration} \rangle
\end{array}
$$

$$\mathit{Spec} \triangleq \mathit{Init} \land \Box[\mathit{Next}]_{\mathit{vars}} \land \mathrm{WF}_{\mathit{vars}}(\mathit{Next})$$

$$
\begin{array}{l}
\mathit{Order} \triangleq \\
\quad \forall\, t \in \text{DOMAIN } \mathit{proposal} : \\
\qquad \forall\, i \in \text{DOMAIN } \mathit{proposal}[t] : \\
\qquad\quad \land\ \land \mathit{proposal}[t][i].\mathit{phase} = \mathit{Commit} \\
\qquad\qquad\quad \land \mathit{proposal}[t][i].\mathit{state}\ = \mathit{InProgress} \\
\qquad\qquad\quad \Rightarrow \neg \exists\, j \in \text{DOMAIN } \mathit{proposal}[t] : \\
\qquad\qquad\qquad\qquad \land\, j > i \\
\qquad\qquad\qquad\qquad \land \mathit{proposal}[t][j].\mathit{phase} = \mathit{Commit} \\
\qquad\qquad\qquad\qquad \land \mathit{proposal}[t][j].\mathit{state}\ = \mathit{Complete} \\
\qquad\quad \land\ \land \mathit{proposal}[t][i].\mathit{phase} = \mathit{Apply} \\
\qquad\qquad\quad \land \mathit{proposal}[t][i].\mathit{state}\ = \mathit{InProgress} \\
\qquad\qquad\quad \Rightarrow \neg \exists\, j \in \text{DOMAIN } \mathit{proposal}[t] : \\
\qquad\qquad\qquad\qquad \land\, j > i \\
\qquad\qquad\qquad\qquad \land \mathit{proposal}[t][j].\mathit{phase} = \mathit{Apply} \\
\qquad\qquad\qquad\qquad \land \mathit{proposal}[t][j].\mathit{state}\ = \mathit{Complete}
\end{array}
$$

$$
\begin{array}{l}
\mathit{Consistency} \triangleq \\
\quad \forall\, t \in \text{DOMAIN } \mathit{target} : \\
\qquad \text{LET} \\
\qquad\quad \text{Compute the transaction indexes that have been applied to the target} \\
\qquad\quad \mathit{targetIndexes} \triangleq \{ i \in \text{DOMAIN } \mathit{transaction} : \\
\qquad\qquad\qquad\qquad \land\, i \in \text{DOMAIN } \mathit{proposal}[t]
\end{array}
$$

$$
\begin{array}{l}
\land\ proposal[t][i].phase = Apply \\
\land\ proposal[t][i].state\ = Complete \\
\land\ t \in transaction[i].targets \\
\land\ \neg\exists\, j \in \text{DOMAIN}\ transaction : \\
\qquad \land\ j > i \\
\qquad \land\ transaction[j].type = Rollback \\
\qquad \land\ transaction[j].rollback = i \\
\qquad \land\ transaction[j].phase = Apply \\
\qquad \land\ transaction[j].state\ = Complete\}
\end{array}
$$

Compute the set of paths in the target that have been updated by transactions

$$
appliedPaths \triangleq \text{UNION}\ \{\text{DOMAIN}\ proposal[t][i].change.values : i \in targetIndexes\}
$$

Compute the highest index applied to the target for each path

$$
\begin{array}{l}
pathIndexes \triangleq [p \in appliedPaths \mapsto \text{CHOOSE}\ i \in targetIndexes : \\
\qquad \forall\, j \in targetIndexes : \\
\qquad\quad \land\ i \geq j \\
\qquad\quad \land\ p \in \text{DOMAIN}\ proposal[t][i].change.values]
\end{array}
$$

Compute the expected target configuration based on the last indexes applied to the target for each path.

$$
expectedConfig \triangleq [p \in \text{DOMAIN}\ pathIndexes \mapsto proposal[t][pathIndexes[p]].change.values[p]]
$$

IN

$$
target[t] = expectedConfig
$$

$$
\begin{array}{l}
Isolation \triangleq \\
\quad \forall\, i \in \text{DOMAIN}\ transaction : \\
\qquad \land\ \land\ transaction[i].phase = Commit \\
\qquad\quad \land\ transaction[i].state\ = InProgress \\
\qquad\quad \land\ transaction[i].isolation = Serializable \\
\qquad\quad \Rightarrow \neg\exists\, j \in \text{DOMAIN}\ transaction : \\
\qquad\qquad\quad \land\ j > i \\
\qquad\qquad\quad \land\ transaction[j].targets \cap transaction[i].targets \neq \{\} \\
\qquad\qquad\quad \land\ transaction[j].phase = Commit \\
\qquad \land\ \land\ transaction[i].phase = Apply \\
\qquad\quad \land\ transaction[i].state\ = InProgress \\
\qquad\quad \land\ transaction[i].isolation = Serializable \\
\qquad\quad \Rightarrow \neg\exists\, j \in \text{DOMAIN}\ transaction : \\
\qquad\qquad\quad \land\ j > i \\
\qquad\qquad\quad \land\ transaction[j].targets \cap transaction[i].targets \neq \{\} \\
\qquad\qquad\quad \land\ transaction[j].phase = Apply
\end{array}
$$

$$
Safety \triangleq \Box(Order \land Consistency \land Isolation)
$$

THEOREM $Spec \Rightarrow Safety$

$$
\begin{array}{l}
Terminated(i) \triangleq \\
\quad \land\ i \in \text{DOMAIN}\ transaction \\
\quad \land\ transaction[i].phase \in \{Apply,\ Abort\}
\end{array}
$$

$\land\ transaction[i].state\ = Complete$

$Termination \triangleq$
$\forall\, i \in 1\,..\,Len(transaction) : Terminated(i)$

$Liveness \triangleq \Diamond Termination$

THEOREM $Spec \Rightarrow Liveness$

Type assumptions.

ASSUME $Nil \in$ STRING

ASSUME $\forall\, phase \in Phase : phase \in$ STRING

ASSUME $\forall\, state\ \in State\ : state\ \in$ STRING

ASSUME $\forall\, status \in Status : status \in$ STRING

ASSUME $\land\, IsFiniteSet(Node)$
$\land\, \forall\, n \in Node :$
$\land\, n \notin$ DOMAIN $Target$
$\land\, n \in$ STRING

ASSUME $\land\, \forall\, t \in$ DOMAIN $Target :$
$\land\, t \notin Node$
$\land\, t \in$ STRING
$\land\, Target[t].persistent \in$ BOOLEAN
$\land\, \forall\, p \in$ DOMAIN $Target[t].values :$
$IsFiniteSet(\,Target[t].values[p])$

---

* Modification History
* Last modified Sun *Feb* 20 08:03:14 *PST* 2022 by *jordanhalterman*
* Created *Wed Sep* 22 13:22:32 *PDT* 2021 by *jordanhalterman*