

Network Services Interface CS State Machine (Utrecht v2)

Two phase reserve and modify

Henrik Thostrup Jensen, htj@nordu.net

Chin Guok, chin@es.net

Tomohiro Kudoh, t.kudoh@aist.go.jp

John MacAuley, john@surfnet.nl

2nd July 2012

Terminology

Terminology - Operations

Abbreviation	Description
rsv (reserve)	The RA requests the PA to reserve network resources for a connection between two STP's constrained by certain service parameters. These resources are held temporarily (commit_timeout) and must be confirmed with a reserveCommit operation.
rsvCommit (reserveCommit)	The reserveCommit operation will commit a previously allocated resources for a connection.
mod (modify)	The modify operation allows a connection reservation to be modified. The resources associated with the modify are only held temporarily (commit_timeout) and must be confirmed with a modifyCommit operation.
modCommit (modifyCommit)	The modifyCommit operation will commit a pending modified connection.
prov (provision)	The RA requests the PA to provision a previously committed reservation.
rel (release)	The RA request for the PA to de-provision resources without removing the reservation.
term (terminate)	The RA request for the PA to release the provisioned resources and terminate the reservation.
query	Mechanism for either RA or PA to query the other NSA for a set of connection service instances between the RA-PA pair. This operation can be used as a status polling mechanism.

Terminology - Messages

Abbreviation	Description
rq (request)	The RA sends the request to the PA, for example reserveRequest.
cf (confirmed)	A PA sends this positive operation response message (such as reserveConfirmed) to the Requester NSA that issued the original request message (reserveRequest).
fl (failed)	A Provider NSA sends this negative operation response message (such as reserveFailed) to the Requester NSA that issued the original request message (reserveRequest).
nt (notification)	A Provider NSA can send an unsolicited messages to the RA (or notification) to communicate to the RA a local event in the PA that resulted in an autonomous state transition in the state machine. An example of this is the “activate_ok.nt” and “activate_ng.nt” notify messages sent from the PA to the RA to indicate a success or failure of the circuit setup in the PA.

Terminology - Notifications

Abbreviation	Description
fcd_end (forcedEnd)	This notification is reported by the PA to the RA to notify that the PA has forced a termination of the reservation.
modify_end	This notification is generated when an NSA has triggered the commit_timeout for a pending modify operation, or if the modify operation was administratively overridden.
activate_ok	This notification is reported by the PA to the RA to notify that the PA has successfully activated the network resources associated with a reservation.
activate_ng	This notification is reported by the PA to the RA to notify that the PA has failed to activate the network resources associated with a reservation.

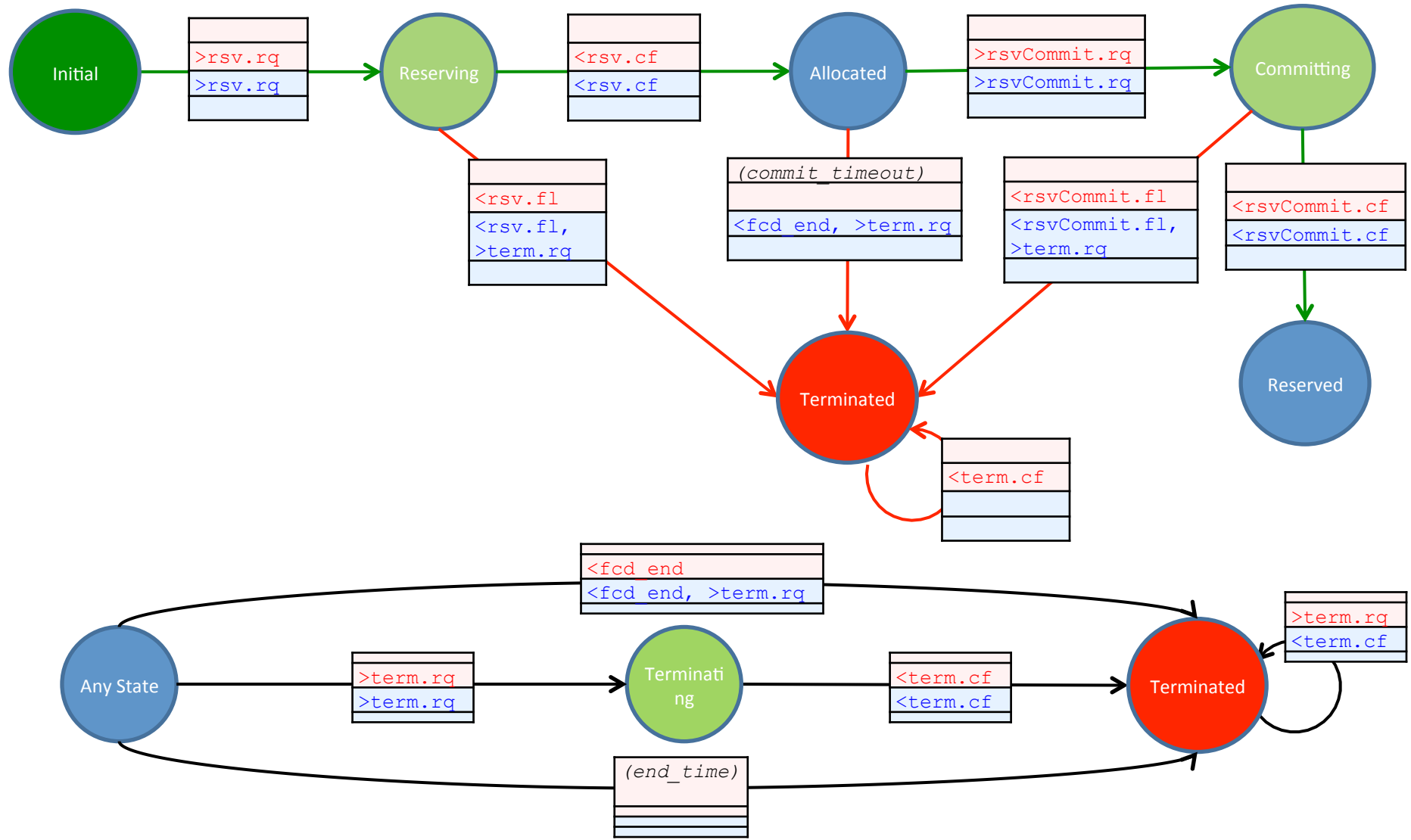
Terminology – NRM operations/events

Abbreviation	Description
(start_time)	The internal NRM event associated with the start time of a connection reservation.
(end_time)	The internal NRM event associated with the end time of a connection reservation.
(commit_timeout)	The internal NRM event associated with the commit message timeout for either the reserveCommit or modifyCommit. If the associated commit message is not received by this time then the reserve or modify message exchange is failed.
(reserve)	The local NRM must perform an internal reserve operation.
(reserve_ok)	The result of the local NRM reserve operation was successful.
(reserve_fl)	The result of the local NRM reserve operation was a failure.
(commit)	The local NRM must perform an internal commit operation for either a pending reserve or modify operation.
(commit_ok)	The result of the local NRM commit operation was successful.
(commit_fl)	The result of the local NRM commit operation was a failure.
(clean_up)	The reservation was terminated and the local NRM must clean up any associated resources.

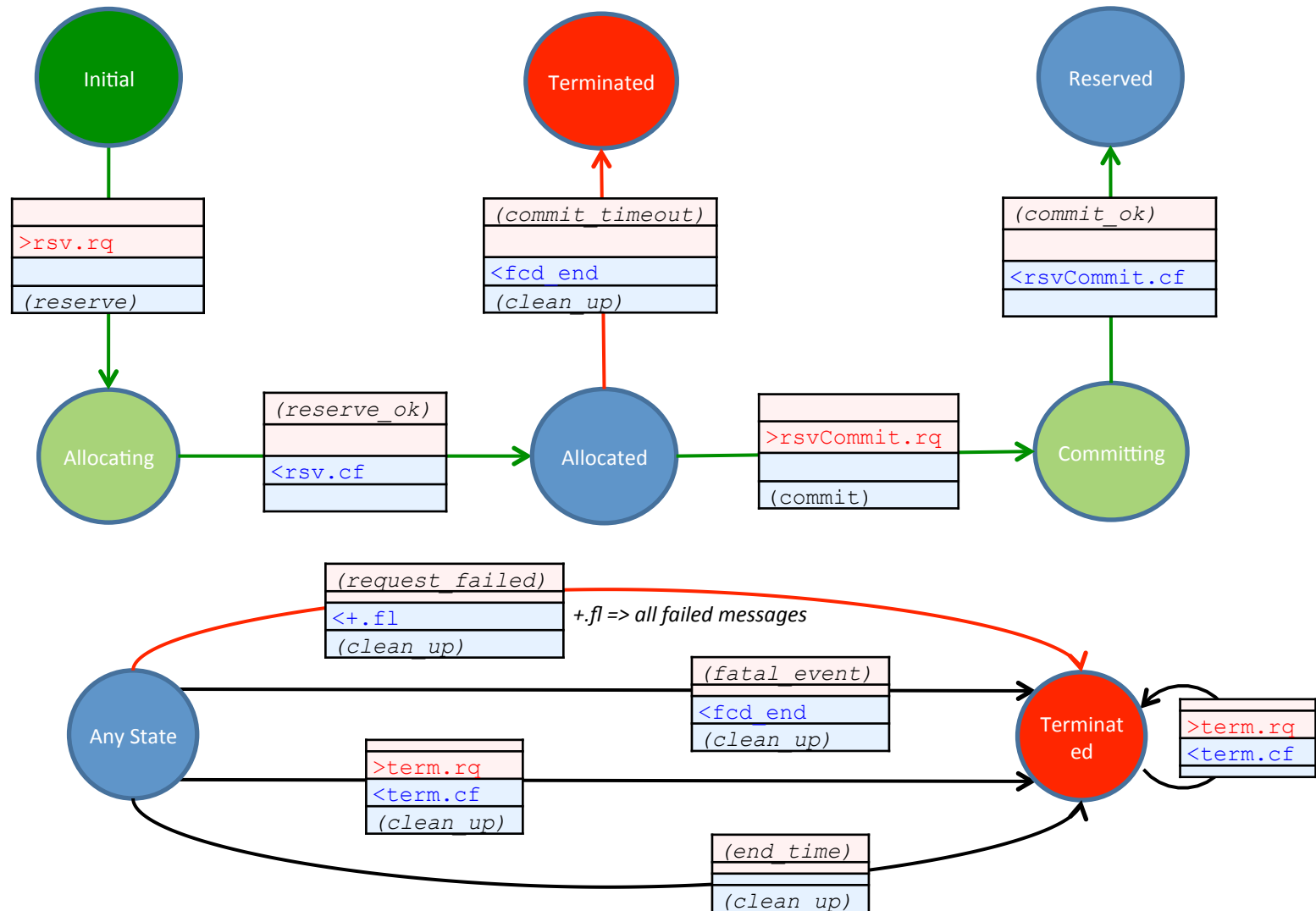
Two-Phase Reserve

Reserving State Machine – uRA/Aggregator

Two Phase



Reservation State Machine – uPA – Two Phase



● Initial State
 ● Transitional States
 ● Stable States
 ● Final State

NB: Requests received in this state is queued and processed only when it transitions to a Stable State.

Reservation – Phase One

- A *reserveRequest* is issued down the tree to check the feasibility of the desired reservation, and to reserve any network resources associated with the request, and a `commit_timeout` timer is started.
- A *reserveConfirm* message will be sent back to the requester if the requested reservation is possible, and as an acknowledgment to successfully securing any required network resources.
- A *reserveFailed* message will be sent back to the requester if the requested reservation is not possible, and the reservation state machine is terminated.
- On successful completion of phase one, any resources associated with the requested reservation are pre-allocated on all participating NSA and will be held until the commit timer expires, or until a *reserveCommit* operation is received.
- If the first phase ends in failure then the reservation is *terminated*.

Reservation – Phase Two

- A *reserveCommitRequest* is issued down the tree to commit the associated reservation.
- A *reserveCommitConfirm* message will be sent back to the requester if the requested reservation is committed successfully and the schedule is in effect.
- A *reserveCommitFailed* message will be sent back to the requester if the reservation commit is not possible, resulting in the termination of the schedule.
- At successful completion of the second phase the reservation has been committed and the network resources allocated.
- If the second phase ends in a failure then a critical error has occurred and the reservation is **terminated**.

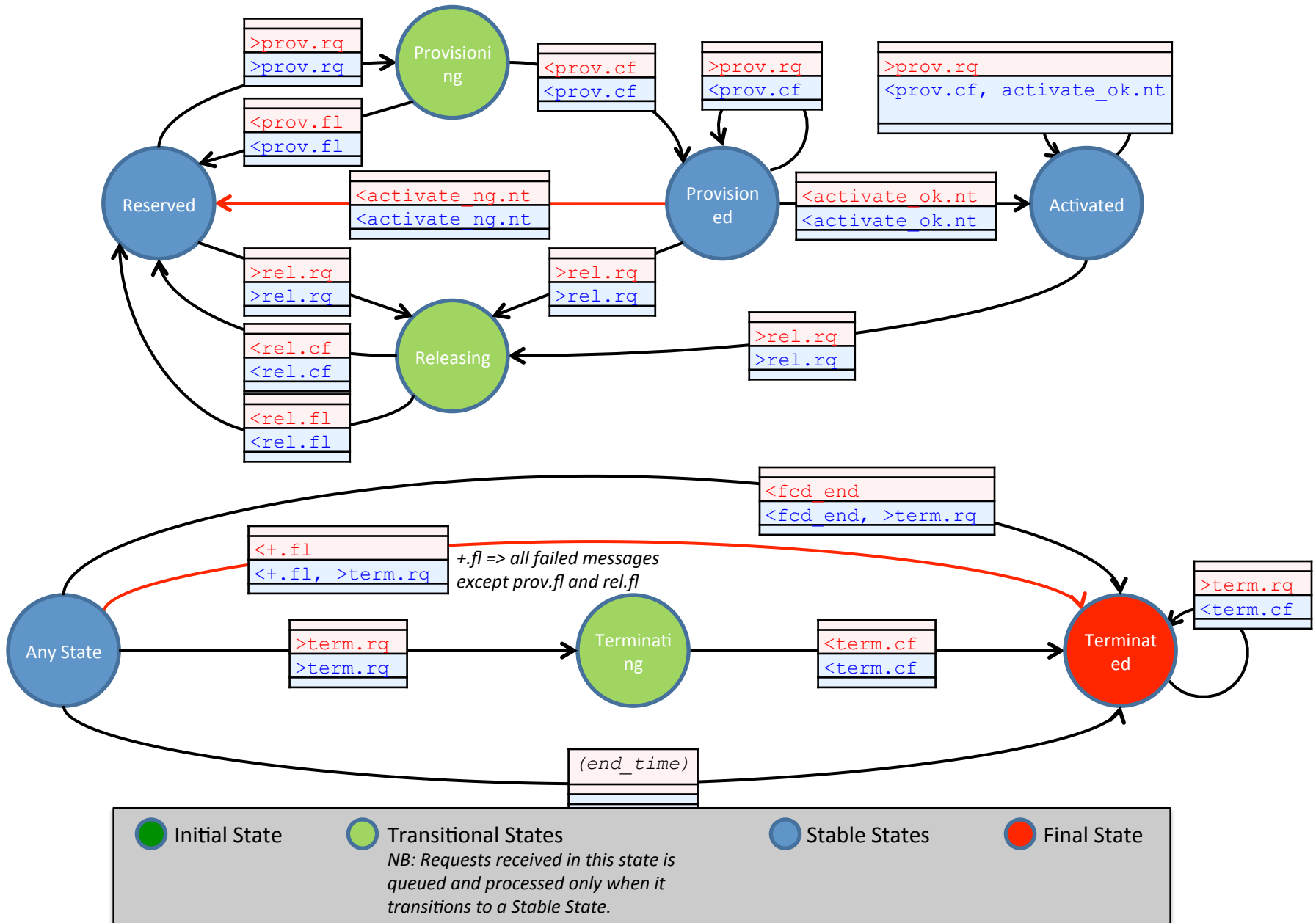
Reserve Commit Timeout

(commit_timeout)

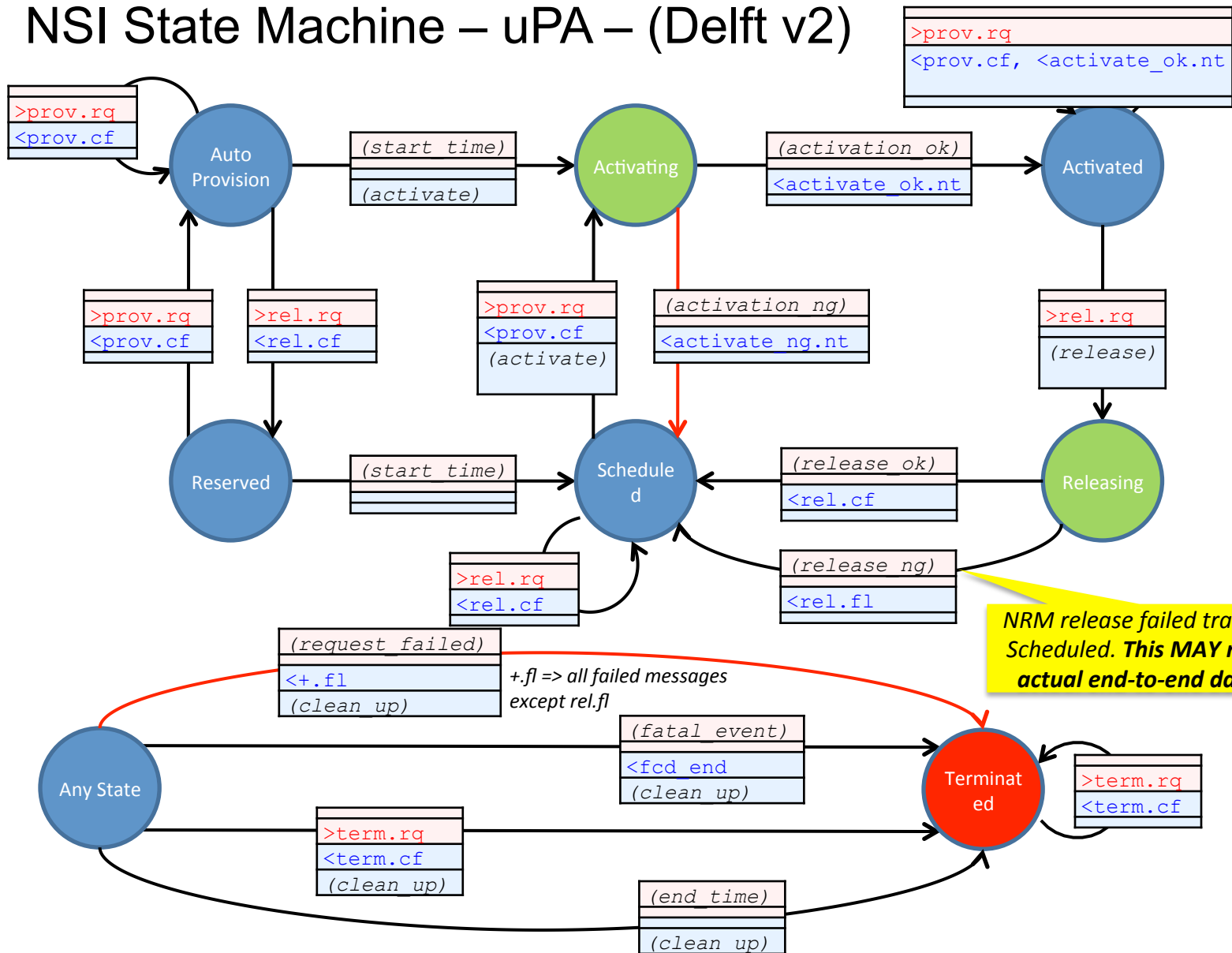
- To avoid leakage of uncommitted reservation resources we implement a commit timeout with a default of **10 minutes**.
- If a corresponding **reverseCommitRequest** is not received within **10 minutes** of a **reserveConfirmed** message then any pending resources against that reservation must be released, and the reservation transitions into a **terminated** state.

Provision/Activate/Release





NSI State Machine – uRA/Aggregator - (Utrecht v1)



NSI State Machine – uPA – (Delft v2)



NRM release failed transitions to Scheduled. This MAY not reflect actual end-to-end dataplane.

 Initial State	 Transitional States	 Stable States	 Final State
<p>NB: Requests received in this state is queued and processed only when it transitions to a Stable State.</p>			

Two-Phase Modify

Modify– Phase One

- A *modifyRequest* is issued down the tree to check the feasibility of the desired reservation modification, and to reserve any network resources associated with the request.
- A `commit_timeout` timer is started.
- A *modifyConfirm* message will be sent back to the requester if the requested reservation modification is possible, and as an acknowledgment to successfully securing any required network resources.
- A *modifiedFailed* message will be sent back to the requester if the requested reservation modification is not possible, and the reservation is returned to its original pre-modification state.
- On successful completion of phase one the original reservation is still preserved, however, any resources associated with the requested reservation change are pre-allocated on all participating NSA and will be held until the commit timer expires, or until a *modifyCancel* or a *modifyCommit* operation is received.
- If the first phase ends in failure then the original reservation is preserved, and will remain in its pre-modification state.

Modify Commit Timeout

(commit_timeout)

- To avoid leakage of uncommitted modify resources we implement a commit timeout with a default of **10 minutes**.
- If a corresponding **modifyCommitRequest** is not received within **10 minutes** of a **modifyConfirmed** message then any pending resources against that reservation must be released, and the reservation remains in the pre-modified state.

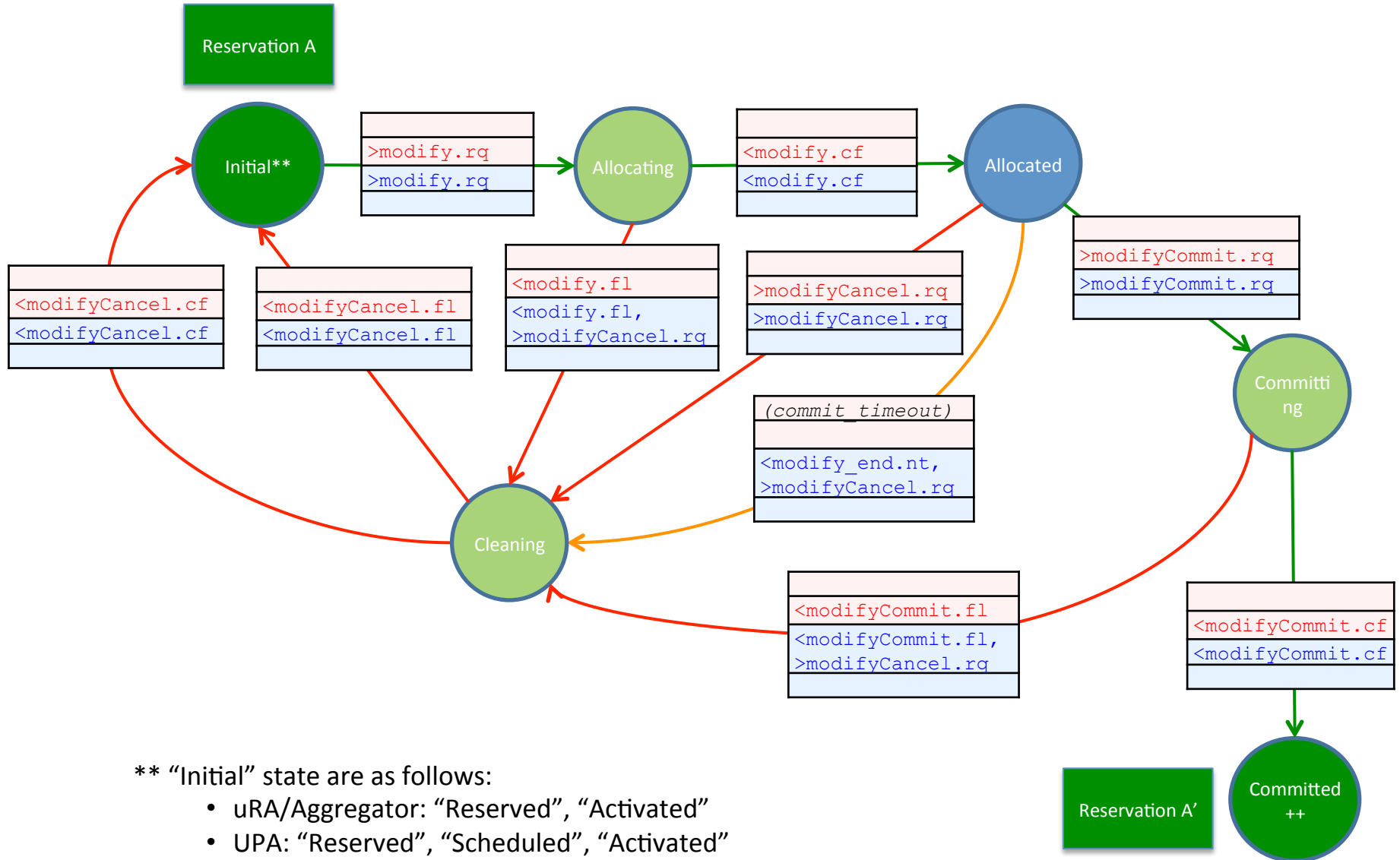
Phase Two

- A *modifyCommitRequest* is issued down the tree to commit the reservation modification.
- A *modifyCommitConfirm* message will be sent back to the requester if the requested modification commit is successful, and the modified schedule is in effect.
- A *modifyCommitFailed* message will be sent back to the requester if the modification commit is not possible (would not be due to lack of resources).
- At successful completion of the second phase the original reservation has been replaced with the modified version.
- If the second phase ends in failure then a critical error has occurred and the reservation is in an indeterminate state within the network.

Canceling a Modification Change

- A modification can be backed out either through the *commit_timeout* occurring, or if a *modifyCancelRequest* is issued down the tree to stop the reservation modification.
- A *modifyCancelConfirm* message will be sent back to the requester if cancellation of the requested modification is successful.
- A *modifyCancelFailed* message will be sent back to the requester if the modification cancel is not possible (improper state or communication error).
- A cancel can only be issued against a reservation if there is a pending modification on the reservation that has not already been committed, or is in the process of committing.

Two Phase Modify - uRA/Aggregator - (Utrecht v1)



** "Initial" state are as follows:

- uRA/Aggregator: "Reserved", "Activated"
- UPA: "Reserved", "Scheduled", "Activated"

++ "Committed" state MUST return to the initial state when modify request was issued.

Two Phase Modify – uPA - (Utrecht v1)

