Network Services Interface

Modify operation proposal

John MacAuley, SURFnet

11th April 2012

What is the problem?

NSI needs the ability to modify an existing connection reservation.

Three key features:

1. Modification of schedule endTime.
2. Modification of reservation bandwidth.
3. Hitless modification of provisioned service.

Notes:

– Modify request should succeed if the existing reservation path can support the request, and fail it otherwise.
– If the existing path can be expanded to allow more bandwidth, or the endTime can be modified without impacting any other schedules, then we should allow it.
– NOT requesting rerouting, bridge and switch, or any other advanced capabilities.

Use Case: Cloud Bypass

SARA has implemented an automatic cloud bypass solution used to reroute huge data transfers between GRID sites

How it works
- Monitor uplink of a switch looking for certain traffic patterns
- When traffic match encountered setup a dynamic lightpath between source and destination belonging to that specific traffic pattern
- Reroute matching traffic over the dynamic path

Why modification?
- Do not know how long the traffic will last, or if the traffic will increase or decrease over time
- Do not want to make a reservation for a week at full capacity of the port
- Make a reservation for an hour and check the traffic pattern again 5 minutes before end of reservation, changing capacity or duration if needed

Modify State Machine – Fire and Pray
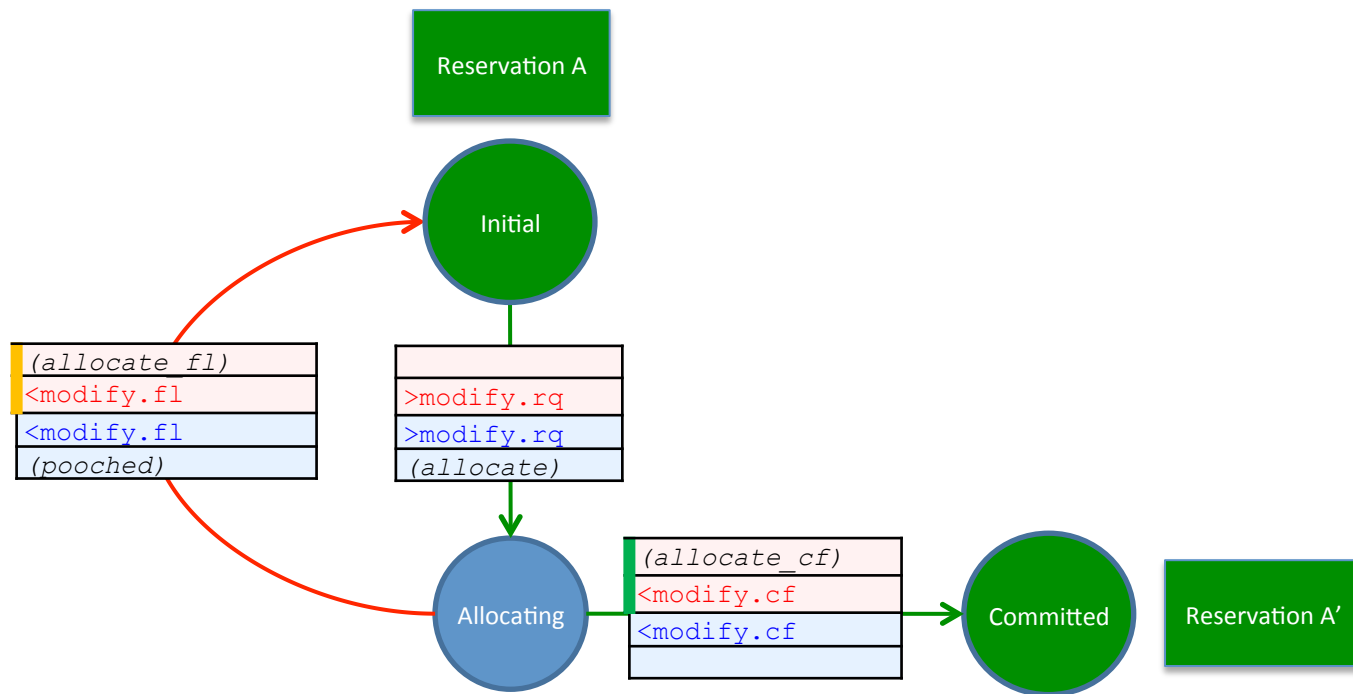
Operations
modify

Issues
In a tree model a child NSA may fail to perform the modify, however, all other children NSA may have committed the change. We now have a reservation within the network that is in an inconsistent state. Backing out the change would be extremely complex, as the requesting NSA would need to remember the original reservation parameters and re-issue them down the tree with another modify to try and restore consistency.

How do we fix this consistency issue?

- In our distributed environment we will need to make modify a two-phase commit operation
- In the first phase…
  - A *modifyRequest* is issued down the tree to check the feasibility of the desired reservation modification, and to reserve any additional network resources associated with the request
  - A *modifyConfirm* message will be sent back to the requester if the requested modification is possible, and as an acknowledgment to successfully securing any additional network resources
  - A *modifyFailed* message will be sent back to the requester if the requested modification is not possible
  - At successful completion of the first phase the original reservation is still preserved, however, pre-allocation of any additional resources associated with the modify has been completed for all participating NSA
  - If the first phase ends in failure then the original reservation is preserved
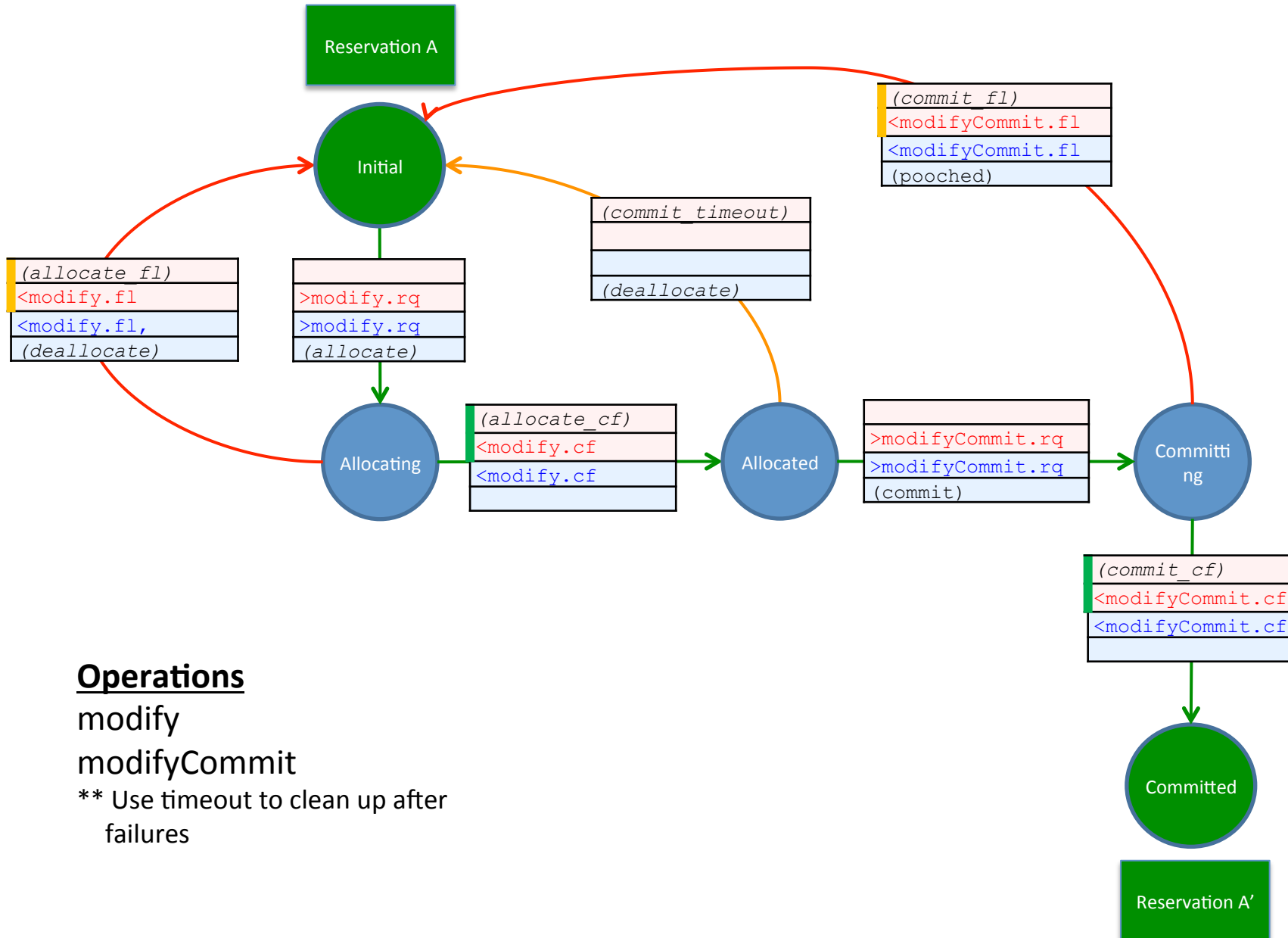
Phase Two

- In the second phase...
  - 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

Commit Timeout

- To avoid leakage of uncommitted modification resources we will need to implement a commit timeout

- If a corresponding *modifyCommitRequest* is not received within 5 minutes of a *modifyConfirmed* message then any pending resources against that reservation should be released

- The original reservation is preserved
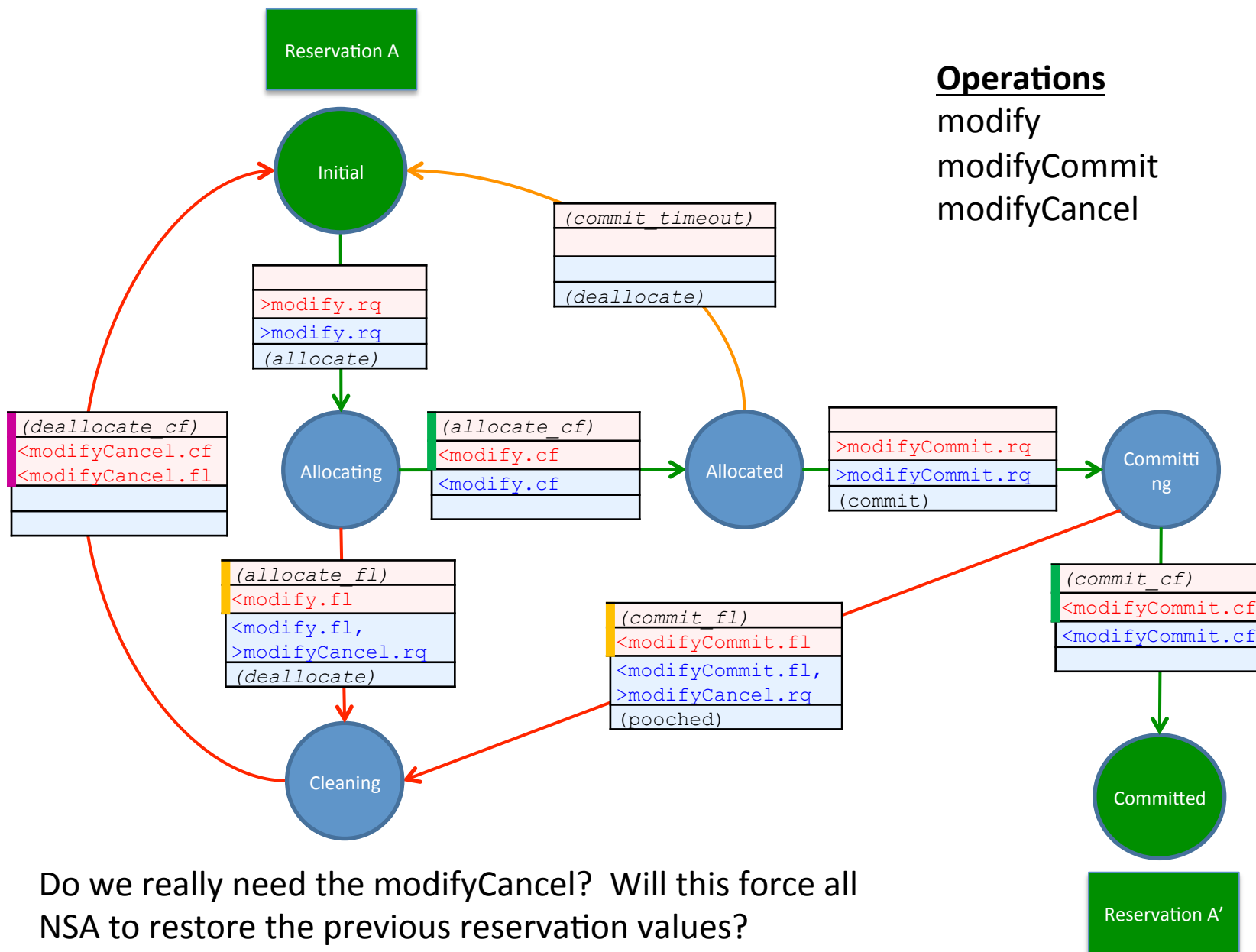
Modify State Machine – Two Phase**

Operations

modify

modifyCommit

** Use timeout to clean up after failures

Modify State Machine – Two Phase

Operations
modify
modifyCommit
modifyCancel

Reservation A

Initial

(commit_timeout)

(deallocate)

>modify.rq
>modify.rq
(allocate)

(deallocate_cf)
<modifyCancel.cf
<modifyCancel.fl

Allocating

(allocate_cf)
<modify.cf
<modify.cf

Allocated

>modifyCommit.rq
>modifyCommit.rq
(commit)

Committing

(allocate_fl)
<modify.fl
<modify.fl,
>modifyCancel.rq
(deallocate)

(commit_fl)
<modifyCommit.fl
<modifyCommit.fl,
>modifyCancel.rq
(pooched)

(commit_cf)
<modifyCommit.cf
<modifyCommit.cf

Cleaning

Committed

Reservation A'

Do we really need the modifyCancel? Will this force all NSA to restore the previous reservation values?

9