- RA can timeout before MTL timeouts. Then RA can cancel the request.
- Message handler can retry, but it is not a requirement.
- after message delivery failure, RA can do either direct retry, or terminte.
- fcd_end is generated by NRM and sent up as a notification to uRA by MH. Therefore there are no fcd_end in the SMs.
- uPA may transit to terminated state when it issues fcd_end. But the uPA must be still able to receive term.rq and send back term.cf.
- retry message uses original request message's correlation id
- after modifying operations, if a NSA is already in RESERVED state, it can receive mdfycncl.rq and reply mdfycncl.cf, but the modification is not rolled back. The system may be in an inconsistent state (different versions across the system) after those operations.
- a NSI_term-rq will cause the RSM, PSM and ASM to be deleted by MH, and further request handling will be done by LSM

con: connection_id (sent by the parent)
cor: correlation_id (sent by the parent)

c_con_i: child connection_id (for child i)
c_cor_i: child correlation_id (for child i)

o_cor: original correlation_id of the request message to retry

---

1. Aggregator MH behavior

NSI messages:
* downstream NSI messages:
- NSI_res.rq(con, cor)

```
if (path finding has succeeded)
{
        create a list of children: CL(con)
        create a set of state machines: RSM(con), PSM(con), LSM(con)
        send res.rq(cor) to RSM(con)
}
else
```

send NSI_res.fl to the parent.

- NSI_mdfychk.rq(con, cor, ver)
send mdfychk.rq(cor, ver) to RSM(con)

- NSI_modify.rq(con, cor, ver)
send modify.rq(cor, ver) to RSM(con)

- NSI_mdfycncl.rq(con, cor, ver)
send mdfycncl.rq(cor, ver) to RSM(con)

- NSI_prov.rq(con, cor)
send prov.rq(cor) to PSM(con)

- NSI_rel.rq(con, cor)
send rel.rq(cor) to PSM(con)

- NSI_retryMessage.rq(con, cor, o_cor) /* NSI_retryMessage includes all the parameters of the original message */
if all flags in RCV(o_cor) are set
send NSI_retryMessage.cf to the parent
else
send NSI_retryMessage.rq to all children in CL(con) which RCV(o_cor) is not set.

- NSI_term.rq
send term.rq(cor) to LSM(con)
send term.rq to RSM(con), PSM(con) /* if RSM and PSM exist */

* upstream NSI messages from children

- NSI_res.cf(c_con, c_cor)
set RCV(c_cor)
if (all of RCV(1..n) are set and FAIL_FLAG(cor)==0)
send res.cf(cor) to RSM(con)

- NSI_res.fl(c_con, c_cor)

set RCV(c_cor)

if(FAIL_FLAG(cor) == 0)

FAIL_FLAG(cor) = 1

send res.fl(cor) to RSM(con)

- NSI_mdfychk.cf(c_con, c_cor, c_ver)

set RCV(c_cor)

if (all of RCV(1..n) are set and FAIL_FLAG(cor)==0)

send mdfychk.cf(cor, ver) to RSM(con)

- NSI_mdfychk.fl(c_con, c_cor, c_ver)

set RCV(c_cor)

if(FAIL_FLAG(cor) == 0)

FAIL_FLAG(cor) = 1

send mdfychk.fl(cor, ver) to RSM(con)

- NSI_modify.cf(c_con, c_cor, c_ver)

set RCV(c_cor)

if (all of RCV(1..n) are set)

send modify.cf(cor, ver) to RSM(con)

- NSI_mdfycncl.cf(c_con, c_cor, c_ver)

set RCV(c_cor)

if (all of RCV(1..n) are set)

send mdfycncl.cf(cor, ver) to RSM(con)

- NSI_prov.cf(c_con, c_cor)

set RCV(c_cor)

if (all of RCV(1..n) are set)

send prov.cf(cor) to PSM(con)

- NSI_rel.cf(c_con, c_cor)

set RCV(c_cor)

if (all of RCV(1..n) are set)

send rel.cf(cor) to PSM(con)

- NSI_term.cf(c_con, c_cor)
    set RCV(c_cor)
    if (all of RCV(1..n) are set)
        send term.cf(cor) to LSM(con)

Input from RSM(con):

- res.rq(cor)
    create RCV_LIST(cor) and FAIL_FLAG(cor)
    send NSI_res.rq(c_con, c_cor) to children in CL(con)

- mdfychk.rq(cor, ver)
    create RCV_LIST(cor)and FAIL_FLAG(cor)
    send NSI_mdfychk.rq(c_con, c_cor, c_ver) to children in CL(con)

- modify.rq(cor, ver)
    create RCV_LIST(cor)
    send NSI_modify.rq(c_con, c_cor, c_ver) to children in CL(con)

- mdfycncl.rq(cor, ver)
    create RCV_LIST(cor)
    send NSI_mdfycncl.rq(c_con, c_cor, c_ver) to children in CL(con)

- res.cf(cor)
    send NSI_res.cf(con, cor) to the parent

- res.fl(cor)
    send NSI_res.fl(con, cor) to the parent

- mdfychk.cf(cor, ver)
    send NSI_mdfychk.cf(con, cor, ver) to the parent

- mdfychk.fl(cor, ver)
    send NSI_mdfychk.fl(con, cor, ver) to the parent

- modify.cf(cor, ver)

send NSI_modify.cf(con, cor, ver) to the parent

- mdfycncl.cf(cor, ver)

send NSI_mdfycncl.cf(con, cor, ver) to the parent

Input from PSM(con):

- prov.rq(cor)

create RCV_LIST(cor)

send NSI_prov.rq(c_con, c_cor) to children in CL(con)

- rel.rq(cor)

create RCV_LIST(cor)

send NSI_rel.rq(c_con, c_cor) to children in CL(con)

- prov.cf(cor)

send NSI_prov.cf(con, cor) to the parent

- rel.cf(cor)

send NSI_rel.cf(con, cor) to the parent

Input from LSM(con)

- term.rq(cor)

create RCV_LIST(cor)

send NSI_term.rq(c_con, c_cor) to children in CL(con)

- term.cf(cor)

clean up everything related to con

send NSI_term.cf(con, cor) to the parent

(TBD)

NSI_activateFailed.nt

NSI_activate Complete.nt

NSI_deactivateComplete.nt

NSI_deactivateFailed.nt

NSI_dataplaneError.nt

NSI_messageNonReach.nt

---

2. uPA MH behavior

NSI messages:

* downstream NSI messages:

- NSI_res.rq(con, cor)

    create a set of state machines: RSM(con), PSM(con), ASM(con), LSM(con)

    send res.rq(cor) to RSM(con)

    if reservation is made by checking the Reservation DB

        send res.cf(cor) to RSM(con)

    else

        send res.fl(cor) to RSM(con)

- NSI_mdfychk.rq(con, cor, ver)

    send mdfychk.rq(cor, ver) to RSM(con)

    if modification is made by checking the Reservation DB

        send mdfychk.cf(cor, ver) to RSM(con)

    else

        send mdfychk.fl(cor, ver) to RSM(con)

- NSI_modify.rq(con, cor, ver)

    send modify.rq(cor, ver) to RSM(con)

- NSI_mdfycncl.rq(con, cor, ver)

    send mdfycncl.rq(cor, ver) to RSM(con)

- NSI_prov.rq(con, cor)

    send prov.rq(cor) to PSM(con)

- NSI_rel.rq(con, cor)

    send rel.rq(cor) to PSM(con)

- NSI_retryMessage.rq(con, cor, o_cor) /* NSI_retryMessage includes all the parameters of the original message */

if REPLIED(o_cor) is set

send NSI_retryMessage.cf to the parent

else

process the original request message

- NSI_term.rq(con, cor)

send term.rq(cor) to LSM(con)

send term.rq to RSM(con), PSM(con), ASM(con) /* if RSM, PSM and ASM exist */

Input from RSM(con):

- res.rq(cor)

ignore

- mdfychk.rq(cor, ver)

ignore

- modify.rq(cor, ver)

ignore

- mdfycncl.rq(cor, ver)

ignore

- res.cf(cor)

set REPLIED(cor)

send NSI_res.cf(con, cor) to the parent

- res.fl(cor)

set REPLIED(cor)

send NSI_res.fl(con, cor) to the parent

- mdfychk.cf(cor, ver)

set REPLIED(cor)

send NSI_mdfychk.cf(con, cor, ver) to the parent

- mdfychk.fl(cor, ver)

set REPLIED(cor)

send NSI_mdfychk.fl(con, cor, ver) to the parent

- modify.cf(cor, ver)

commit the reservation(con, ver)

set REPLIED(cor)

send NSI_modify.cf(con, cor, ver) to the parent

- mdfycncl.cf(cor, ver)

abort the reservation(con, ver)

set REPLIED(cor)

send NSI_mdfycncl.cf(con, cor, ver) to the parent

Input from PSM(con):

- prov.rq(cor)

set prov_flag(con)

if in_period_flag is set

send NRM a request to activate data plane according to the latest reservation

send prov.cf(cor) to PSM(con)

- rel.rq(cor)

reset prov_flag(con)

send NRM a request to deactivate data plane;

send rel.cf(cor) to PSM(con)

- prov.cf(cor)

send NSI_prov.cf(con, cor) to the parent

- rel.cf(cor)

send NSI_rel.cf(con, cor) to the parent

Input from LSM(con)

- term.rq(cor)

ignore

- term.cf(cor)

clean up everything related to con

send NSI_term.cf(con, cor) to the parent

(TBD)

Input from ASM(con)

- outOfService.nt
- inService.nt

Input from NRM