

# R2 Recovery Design [Init | LostConnection]

- 0. Agenda
- 1. RNibWriter
  - 1.1. Currently
  - 1.2. Changes
- 2. Lost Connection flow
  - 2.1. Currently
  - 2.2. New Proposed Flow
  - 2.3. Sequence Diagram
- 3. E2Terminator Init flow
  - 3.1. E2TerminatorInitHandler Algorithm
  - 3.2. Sequence Diagram
- 4. Zeroing Connection Attempts
- 5. RanConnectionManager
  - 5.1. ReconnectRan(ranName)
  - 5.2. ExecuteSetup
- 6. Appendix
  - 6.1. ValidateRanConnectionForRecoverySetup implementation suggestion
  - 6.2. Current Setup implementation (triggered from HTTP flow)

## 0. Agenda



This design is dedicated for the purpose of 2Q2019 Recovery epic (in "R2"): RICPLT-1850

## 1. RNibWriter

### 1.1. Currently

These are the current keys we save in *RNib* when triggering *SaveNodeb*:

KEY	VALUE
RAN:<INVENTORY_NAME>	<NodebInfo>
<NODE_TYPE>:<PLMN_ID>:<NB_ID>	<NodebInfo>
CELL:<CELL_ID> OR NRCELL:<NRCELL_ID>	<Cell>
PCI:<INVENTORY_NAME>:<PCI>	<Cell>
ENB	[<NbIdentity>]
GNB	[<NbIdentity>]
UNKNOWN	[<NbIdentity>]

### 1.2. Changes

- ConnectionAttempts
  - Add two new fields to the *NodebInfo* struct (as well as a new enum for determining E2AP):

Error rendering macro 'code': Invalid value specified for parameter 'com.atlassian.confluence.ext.code.render.InvalidValueException'

```
E2ApplicationProtocol e2_application_protocol = 4;
uint32 connection_attempts = 6;
```

```
enum E2ApplicationProtocol {
    UNKNOWN_ASN1_MESSAGE_TYPE = 0;
    X2_SETUP_REQUEST = 1;
    ENDC_X2_SETUP_REQUEST = 2;
}
```

- We shall regenerate pb.go and notice that we will fetch this value for GetNodeb query AND REST as well
- Motivation for adding it to NodebInfo struct:

- Every time we update ConnectionStatus to CONNECTING we shall update ConnectionAttempts as well
- The alternative of a different key/value dedicated for this purpose will result in couple of DB queries (one for updating status, one for updating attempts)
- The alternative of moving connectionStatus from NodebInfo struct will result in making two fetches from DB since we would need to fetch the NodebInfo as well as fetch the status in GET, and according to Architect GET happens A LOT.
- UpdateNodebInfo
  - Add a new method to *RnibWriter*:
  - Error rendering macro 'code': Invalid value specified for parameter 'com.atlassian.confluence.ext.code.render.InvalidValueException' UpdateNodebInfo(nbIdentity \*entities.NbIdentity, nodebInfo \*entities.NodebInfo) error
  - Motivation:
    - This method will update ONLY the RAN:<INVENTORY\_NAME> & <NODE\_TYPE>:<PLMN\_ID>:<NB\_ID> entries with the provided nodebInfo pointer.
    - Using this function allows us to update information without the overhead of building the cells again yet saving the exact information we need.
- Another future change will be **adding a StartTime property to the NodebInfo struct**.  
When this will be done, we could **DROP** the e2sessions map.

## 2. Lost Connection flow

### 2.1. Currently

- E2T sends a RIC\_SCTP\_CONNECTION\_FAILURE message
- E2M executes RanLostConnectionHandler
- E2M queries RNIB using GetNodeb
- E2M switches states according to the following rules:

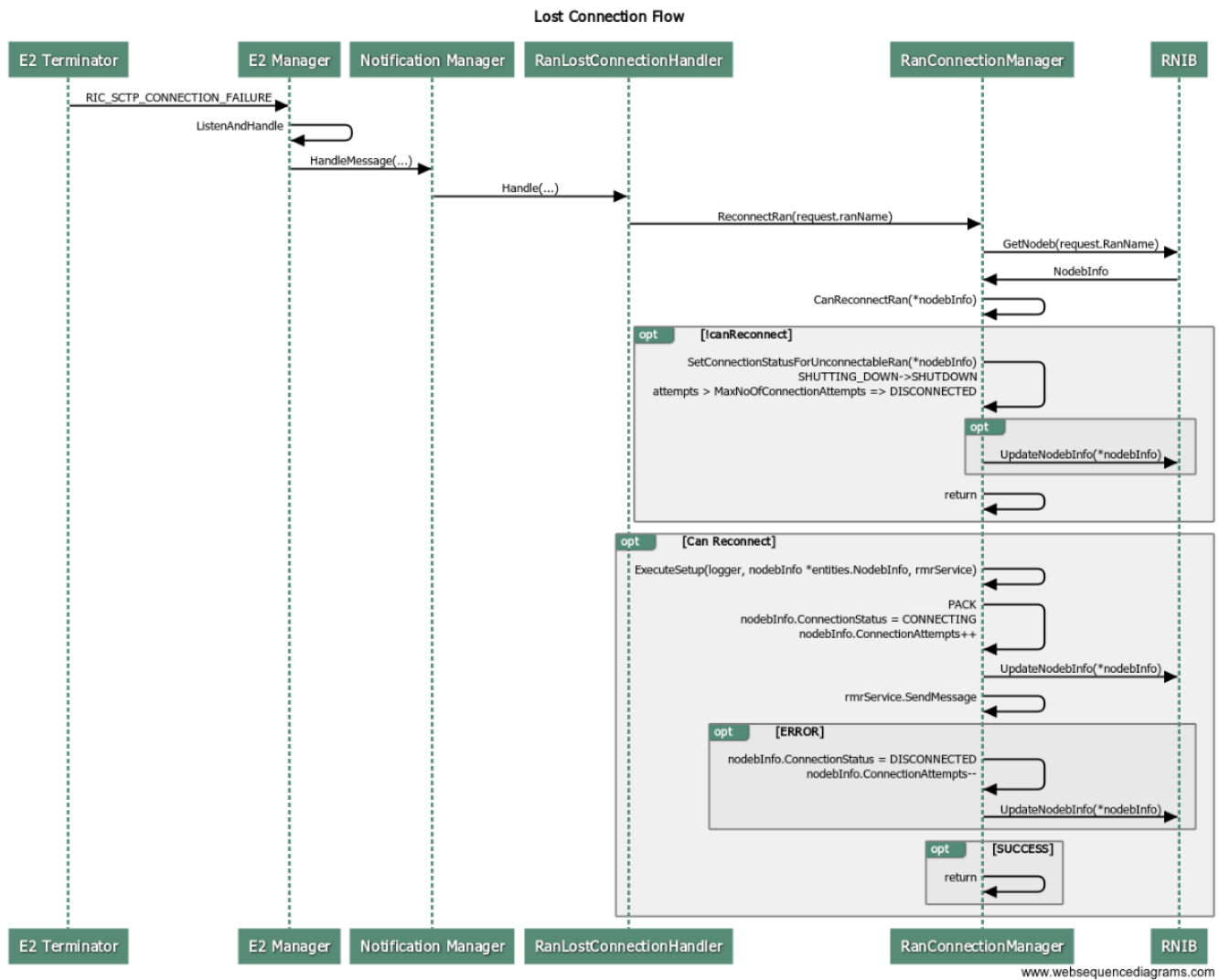
Current State	Next State
CONNECTED    CONNECTING    CONNECTED_SETUP_FAILED	DISCONNECTED
SHUTTING DOWN    SHUT DOWN	SHUT DOWN

- E2M uses SaveNodeb to save the changed state (this function is used also at SHUTDOWN flow)

### 2.2. New Proposed Flow

- Add to configuration MaxNoOfRanConnectionAttempts
- RanLostConnectionHandler: call *RanConnectionManager.ReconnectRan(ranName)* - please review 5.1.

### 2.3. Sequence Diagram



### 3. E2Terminator Init flow

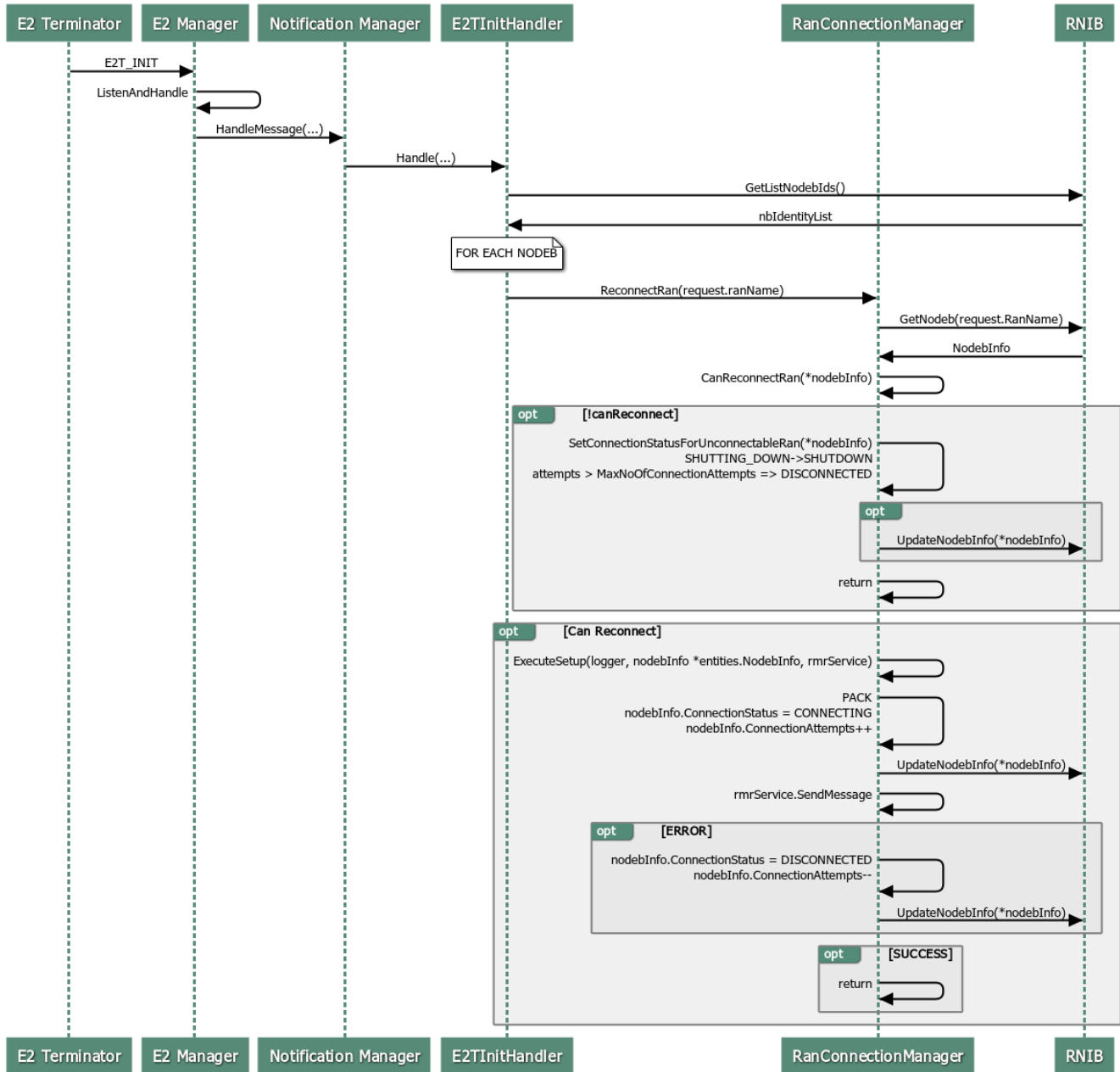
**i** We shall add a new RMR message code in the routing table, as well as attach a dedicated *E2TerminatorInitHandler*.

#### 3.1. E2TerminatorInitHandler Algorithm

- Execute handler.readerProvider().GetListNodebIds().
- Iterate the nodebIdList, and for each Nodeb
  - Call *RanConnectionManager.ReconnectRan(ranName)* - please review 5.1.

#### 3.2. Sequence Diagram

E2T Init Flow



www.websequencediagrams.com

## 4. Zeroing Connection Attempts

We shall reset nodebInfo.ConnectionAttempts in all of the following code zones:

- X2/ENDC HTTP setup requests BEFORE sending to RMR
- X2/ENDC setup response handler
- X2/ENDC setup failure response handler

## 5. RanConnectionManager

### 5.1. ReconnectRan(ranName)

- Query RNIB for the NodeInfo (GetNodeb).

- **ERROR:** Log and return
- **SUCCESS:**

- Execute **CanReconnectRan**(ranConnectionStatus, ranConnectionAttempts)  
A RAN can be reconnected if it fulfills the following conditions:

```
ranConnectionStatus != SHUTTING_DOWN && ranConnectionStatus != SHUT_DOWN &&
ranConnectionAttempts <= MaxNoOfConnectionAttempts
```

- **CANNOT RECONNECT:**

- Execute **SetConnectionStatusForUnconnectableRan**(nodebInfo)

Error rendering macro 'code': Invalid value specified for parameter 'com.atlassian.confluence.ext.code.render.InvalidValueException'

```
if nodeb.ConnectionStatus == SHUTTING_DOWN {
    nodeb.ConnectionStatus = SHUTDOWN
    UpdateNodebInfo(nodeb)
}

if nodeb.ConnectionAttempts > MaxNoOfConnectionAttempts {
    nodeb.ConnectionStatus = DISCONNECTED
    UpdateNodebInfo(nodeb)
}
```

- Log and return

- **READY FOR SETUP:**

- Call **RanSetupManager.ExecuteSetup**(nodebInfo \*entities.NodebInfo, rmrService)

## 5.2. ExecuteSetup

For current implementation, please review 6.2.

```
func ExecuteSetup(rnibWriterProvider func() rNibWriter.RNibWriter, nodebInfo *entities.NodebInfo, rmrService
*services.RmrService, startTime time.Time) error
```

### Algorithm:

1. Pack (RanIp,RanPort,RanName are derived from NodebInfo)
2. Update NodebInfo ConnectionStatus to CONNECTING ,increment ConnectionAttempts and Save to RNIB

```
nodebInfo.ConnectionStatus = CONNECTING
nodebInfo.ConnectionAttempts++
UpdateNodebInfo(nodebInfo)
```

3. Send to RMR using rmrService.SendMessage

- 3.1. **ERROR:** Update NodebInfo ConnectionStatus to DISCONNECTED, decrease ConnectionAttempts and Save to RNIB

```
nodebInfo.ConnectionStatus = DISCONNECTED
nodeBInfo.ConnectionAttempts--
UpdateNodebInfo(nodebInfo)
```

- 3.2. **SUCCESS:** DONE

\* Notice: At the next phase, we won't send startTime as a parameter, we will take it from NodebInfo.

## 6. Appendix

### 6.1. ValidateRanConnectionForRecoverySetup implementation suggestion

Error rendering macro 'code': Invalid value specified for parameter 'com.atlassian.confluence.ext.code.render.InvalidValueException'

```
isValid, nextConnectionStatus := ValidateRanConnectionForRecoverySetup(connectionStatus, connectionAttempts)

if !isValid {
    if nextConnectionStatus != nil {
        UpdateNodebInfo(nextConnectionStatus)
    }
    return
}

ValidateRanConnectionForRecoverySetup(*ConnectionStatus, ConnectionAttempts) (bool, *ConnectionStatus) {
    if connectionStatus == SHUTTING_DOWN {
        return false, ConnectionStatus.SHUT_DOWN
    }

    if connectionStatus == SHUT_DOWN {
        return false, nil
    }

    if connectionAttempts > MaxNoOfRanConnectionAttempts {
        return false, ConnectionStatus.DISCONNECTED
    }

    return true, nil
}
```

## 6.2. Current Setup implementation (triggered from HTTP flow)

Error rendering macro 'code': Invalid value specified for parameter 'com.atlassian.confluence.ext.code.render.InvalidValueException'

```
var wg sync.WaitGroup

go handler.CreateMessage(rc.Logger, &requestDetails, messageChannel, E2Sessions, startTime, wg)

go rc.rmrService.SendMessage(handler.GetMessageType(), messageChannel, errorChannel, wg)

wg.Wait()

err = <-errorChannel

if err != nil {
    handleErrorResponse(rc.Logger, http.StatusInternalServerError, internalErrorCode, sendMessageErrorMessage,
writer, startTime)
    return
}

func (SetupRequestHandler) CreateMessage(logger *logger.Logger, requestDetails *models.RequestDetails,
messageChannel chan *models.E2RequestMessage, e2sessions sessions.E2Sessions, startTime time.Time, wg sync.
WaitGroup)
```