Network Management Overview

- Managing several control units linked together via their network nodes
- Managing different ECU those exchange information via the same CAN bus

Network Management (NM) Task

- Startup and shut down of network communication
- Handling communication failure
- Appointment for Network Configuration
- Inspection of network configuration
- Power management
- Node management

OSEK NM

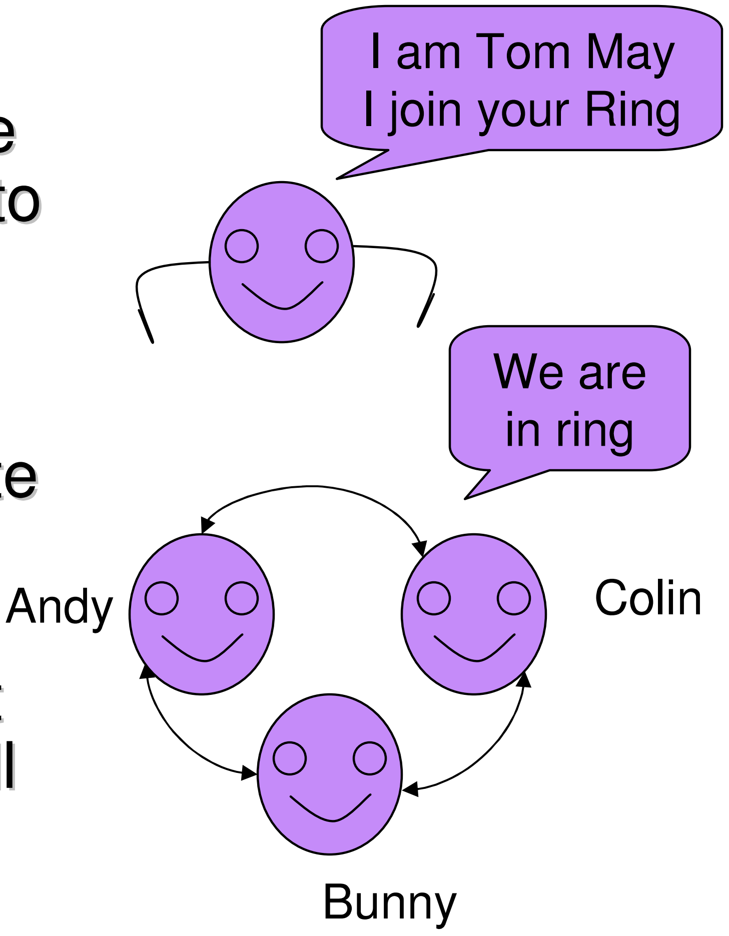
- It is a token based strategy for the communication.
- Besides normal application message a special message is defined for each node
- it passes the network information and control the transmission and reception of application message

OSEK NM Function

- Alive
- Ring
- Sleep
- Wakeup
- Sleep indication
- Limp home
- Sleep acknowledge

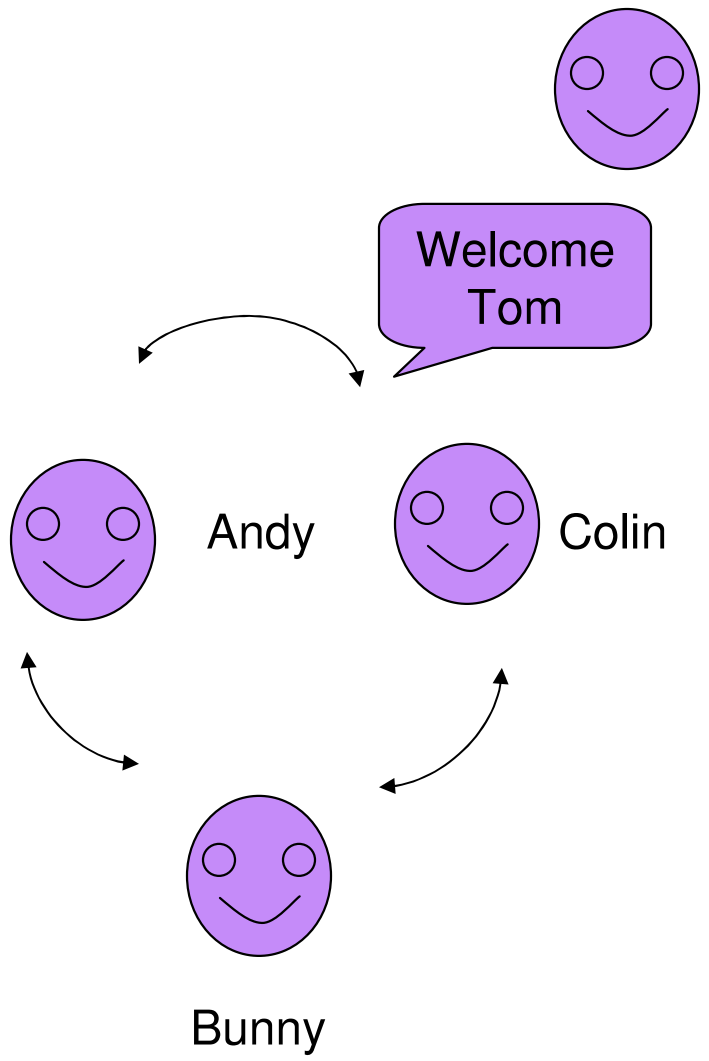
Alive

- If a node needs the network and want to join the communication it sends an “Alive” message to indicate the other nodes of existing
- When a node want to join a ring it shall send its can message with “Alive” bit set

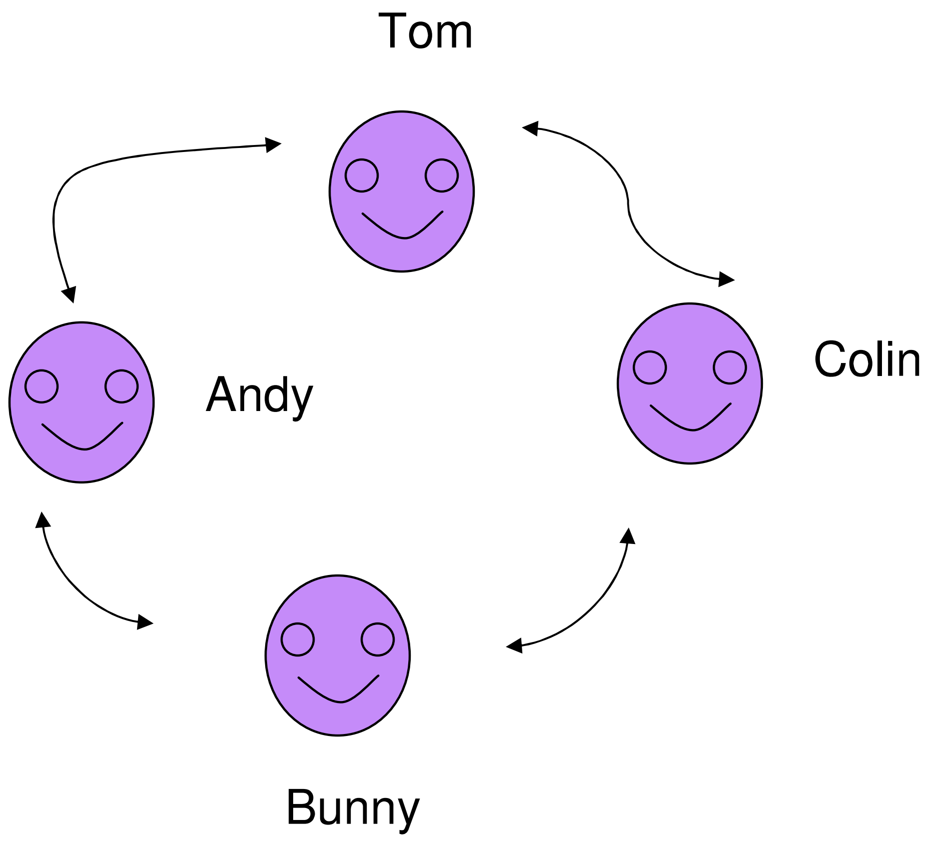


Ring

- Alived ECU will build up logical ring to achieve a network-wide synchronizati on of all ECU nodes that take part in Communicati on



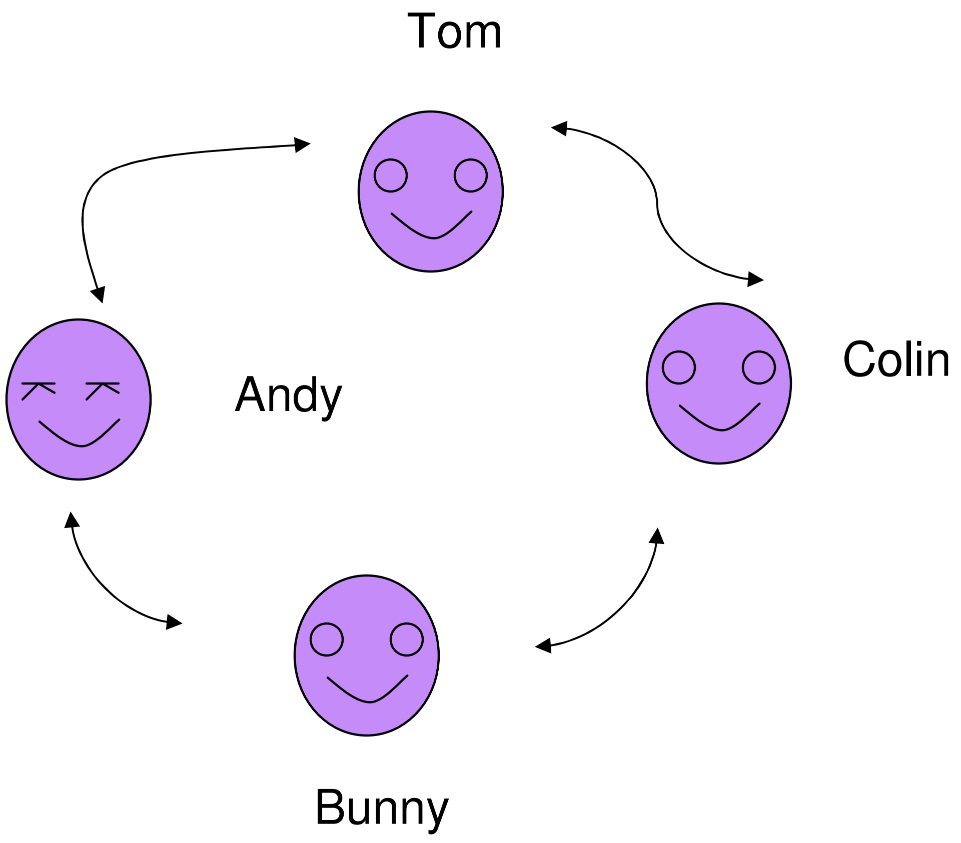
After Ring Formation



Sleep Indication

- When ECU does not need the network it sends out “Sleep Indication” to inform the other ones that it want to off the line

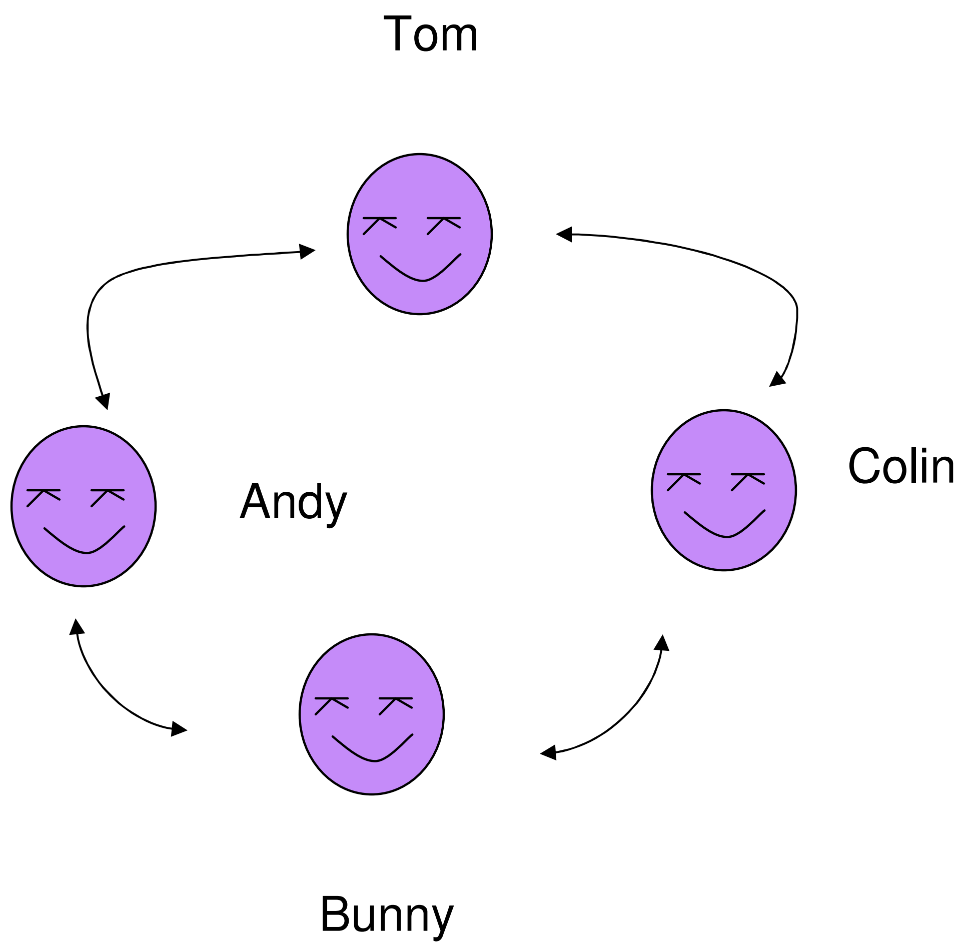
Let us assume Andy wants to sleep



Sleep Acknowledge

- When All ECU sets sleep indication a sleep acknowledge bit will be set

Sleep Acknowledge state



How to implement NM layer in CANOe

- Network management consists of the following two files
  - OSEKNM01.DLL
  - OSEKNM.INI
- They may be stored with the CANoe configuration, e.g. in a directory called .exec32

CAN db Attributes

- The DLL obtains its information from the appended database.
- This information is contained in suitable attributes. The following attributes must be included in the database for Osek network management.
- The attribute name must be entered in the same way as formulated below

CAN db Attributes

- The maximum number N of nodes participating in Network Management can be determined via the attribute **NmMessageCount**.
- When doing this, see to it that the number is always set as 2^M, with M as a natural number (hence N= 16, 32 or 64).
- In turn, the base address of this CAN Identifier area must be divisible by the number of nodes with no remainder.
- The base address is also determined in the database and described with the network attribute **NmBaseAddress**.

CAN db Attributes

- **NMIdType** (enumeration)
- This attribute defines if standard or extended IDs are used. Valid values are:
  - 0: standard (11 bit, default)
  - 1: extended (29 bit)

CAN db Attributes

- **NmCAN -** See network attribute NmCAN. This node attribute allows to set the CAN channel on node level

CAN db Attributes

- **NodeLayerModules** (string) ***mandatory***
- The attribute NodeLayerModules must be defined as a string field and contains the name of the NodeLayer DLL, i.e. **OSEKNM01.DLL**.

Osek Network Management DLL Interface Functions

- API functions which the DLL provides and can be called from CAPL
- The CALLBACK functions which must be defined in CAPL and are called by the DLL.

API Functions Of The DLL

- The Network Management DLL provides 16 different API functions.
- The OSEK names agree with both the OSEK specification and with the name used in the ECUs

TalkNM

- void **TalkNM**(void); [old-style name: OsekNMTalk]
- Network Management is switched to active, so that it can participate in the ring.

SilentNM

- void **SilentNM**(void); [OsekNMSilent]
- After this call Network Management does not participate in the ring any more
- It does not observe incoming messages

GotoMode_BusSleep

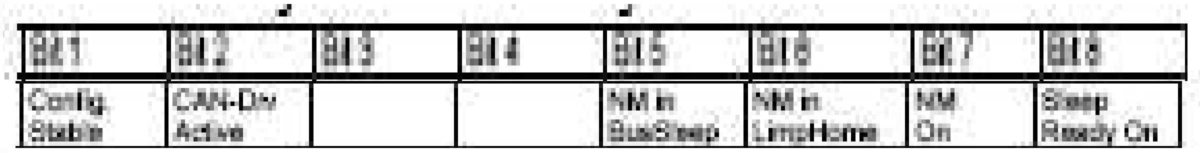
- void **GotoMode_BusSleep**(void); [OsekNMBusSleepInd]
- This function communicates to Network Management that the application is ready to sleep

GotoMode_Awake

- void **GotoMode_Awake**(void); [OsekNMAwake]
- The application can cancel its readiness to sleep with this function while Network Management is running.
- If the sleep state has already been entered, this function must be called to initialize Network Management.
- This function is then generally called by the function ApplCanWakeUp() (cf. CALLBACK Function 4)

NMGetStatus

- unsigned long **NMGetStatus**(void); [OsekNMGetStatus]
- The internal status of Network Management is retrieved with this function



Bit 1	Bit 2	Bit 3	Bit 4	Bit 5	Bit 6	Bit 7	Bit 8
Config. Stable	CAN-Drv Active			NM in BusSleep	NM in LimpHome	NM On	Sleep Ready On

Canonline()

- This function make the node to available and transmit the messages in CAN bus

Canoffline

- This function make the node to unavailable and stop transmitting the messages in CAN bus

Functions Provided by CAPL

- void **apBusSleep**(void)
- void **apCanSleep**(void)
- void **apCanNormal**(void)
- void **apCanWakeUp**(void)
- void **apCanOff**(void)
- void **apCanOn**(void)
- void **apIndRingData**(void)
- void **apIndLastAwake**(void)

void **apBusSleep**(void) (CALLBACK function 1)

- The signal SleepAck was received/sent, and time T_BUSSLEEP has elapsed. From standby mode the CAN transceiver can be placed in sleepmode here.
- In the transition to sleep mode this function is called directly after the function call apCanSleep().

void **apCanSleep**(void) (CALLBACK function 2)

- The signal SleepAck was received/sent, and time T_BUSSLEEP has elapsed. From active mode the CAN transceiver can be placed in standbymode here.
- In the transition to sleep mode this function is called directly before the the function call apCanBusSleep().

void **apCanNormal**(void) (CALLBACK function 3)

- This is called to place the CAN transceiver in the active state. This function is called if the function OsekNMAwake() was called and there was a bus quiet state.

void **apCanWakeUp**(void) (CALLBACK function 4)

- This function is called if the network management DLL has received a message while in the sleep state.
- In this case the function *OsekNMAwake()* must be called to activate network management! If the network management is to remain sleep-ready, the function *OsekNMBusSleepInd()* must be called ***after*** the Awake call

void **apCanOff**(void) (CALLBACK Function 5)

- This function is called during the transition to sleep mode and instructs the CAPL node not to send any more messages.

void **apCanOn**(void)

- Network management is reactivated. CAN output is enabled again.

void **apIndRingData**(void) (CALLBACK-Function 7)

- This function will be called if the node receives the token and the ring is stable.
- The application can get the latest data received by this NM-message.

void **apIndLastAwake**(void)

- This function is called when the node is the last active node on the bus that does not signal sleep readiness. Therefore the bus would go into sleep mode
- if this node also signaled sleep readiness. The condition is checked on the reception of every NM message where the sleep readiness bit is set, therefore this function can be called repeatedly.

OSEKNM.INI

- [LINEMODE]
- Chips = 0; ; 0 - both, 1 - chip 1 only, 2 - chip 2 only
- [WAKEUP]
- WakeUpOnAllMsg = 1
- [Debug]
- ;AcknowledgeRcv = 0
- ;DebugId = 0x222;
- [MESSAGE]
- DlcLen=8 ; Min. 2 max. 8
- RingDataStartIndex=2 ; Min. is 2
- RingDataLength=6 ; max. is 6

OSEKNM.INI

```
RingDataFill=0
; Specify the value to initialize the ring data with
;RingDataInit=0
[WRITE]
VERBOSE = 1 ; verbose level: 0 - no messages, 1 - some messages, 2 - more
messages,
3 - even more messages
[NAMES]
CALLBACKFunc1 = apBusSleep
CALLBACKFunc2 = apCanSleep
CALLBACKFunc3 = apCanNormal
CALLBACKFunc4 = apCanWakeUp
CALLBACKFunc5 = apCanOff
CALLBACKFunc6 = apCanOn
CALLBACKFunc7 = apIndRingData
[TIMER]
T_RING_TYP = 100 ; Typical time interval between two ring messages
T_RING_MAX = 250 ; max. time interval between two ring
T_NOTLAUF = 1000 ; time interval between two ring messages with NMLimpHome
; identification
T_BUSSLEEP = 1500 ; Time the NM waits before transmission into the state
; NMBusSleep
```

Structure of Network Management Messages

- Data byte 1: Receiver address For the messages ALIVE and LIMPHOME the node.s own station address is found here. This station.s address is also entered here for the first RING message, provided that no other NM message could have been received between the ALIVE message and this RING message. Otherwise the address of the logical successor within the ring will be located here.

Structure of Network Management Messages

- Data byte 2: Identifies the type of NM message and contains sleep information
  - Bit 1: ALIVE message
  - Bit 2: RING message
  - Bit 3: LIMPHOME message
  - Bit 4: Reserved
  - Bit 5: Sleep readiness (SleepIndicator)
  - Bit 6: Sleep acknowledgment (SleepAcknowledge)
  - Bit 7: Reserved
  - Bit 8: Reserved