

# Automotive Embedded System II

- OSEK/VDX – COM&NM

Spring 2021

Sukhyun Seo



1. OSEK/VDX COM
  - 1.1. 개요
  - 1.2. 계층 구조
  - 1.3. Interaction Layer
  - 1.4. Message Reception
  - 1.5. Message Transmission
  - 1.6. Communication System Management
  - 1.7 Deadline Monitoring
2. OSEK/VDX NM
  - 2.1. 개요
  - 2.2. Direct OSEK NM
  - 2.3. Indirect OSEK NM

## 1. OSEK/VDX COM

### 1.1. 개요

### 1.2. 계층 구조

### 1.3. Interaction Layer

### 1.4. Message Reception

### 1.5. Message Transmission

### 1.6. Communication System Management

### 1.7 Deadline Monitoring

## 2. OSEK/VDX NM

### 2.1. 개요

### 2.2. Direct OSEK NM

### 2.3. Indirect OSEK NM

- 특징

- ✓ Automotive Control Unit Application Software을 위한 통일된 통신 환경
- ✓ Application Software Modules의 portability를 높이기 위해 Software Communication Interfaces와 ECU 내부 통신 특성, ECU 외부 통신 특성 등을 정의
- ✓ Task와 ISR 사이의 데이터 전송을 위한 Services 제공
- ✓ 특정 Application Program Interface를 통해서만 가능
- ✓ 규격은 Portability, reusability and interoperability를 보장
- ✓ OSEK indirect NM 을 지원하기 위한 Services 제공

## 1. OSEK/VDX COM

### 1.1. 개요

### 1.2. 계층 구조

### 1.3. Interaction Layer

### 1.4. Message Reception

### 1.5. Message Transmission

### 1.6. Communication System Management

### 1.7 Deadline Monitoring

## 2. OSEK/VDX NM

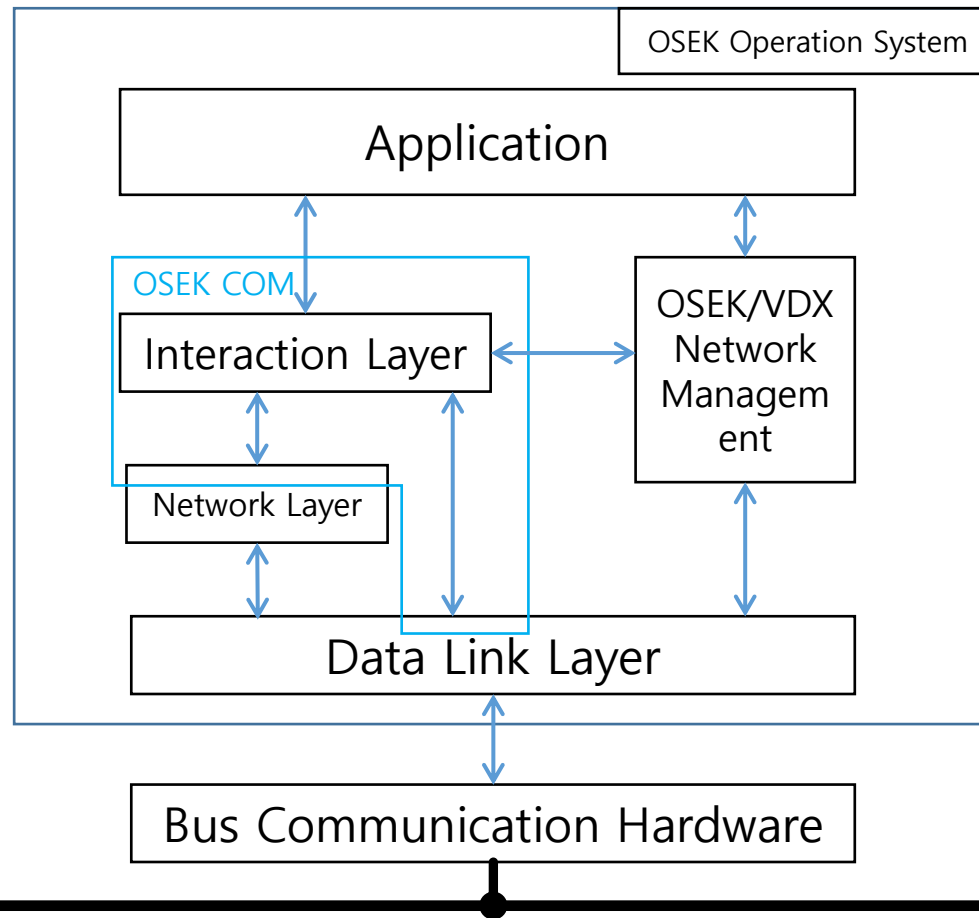
### 2.1. 개요

### 2.2. Direct OSEK NM

### 2.3. Indirect OSEK NM

## 1.2. 계층 구조

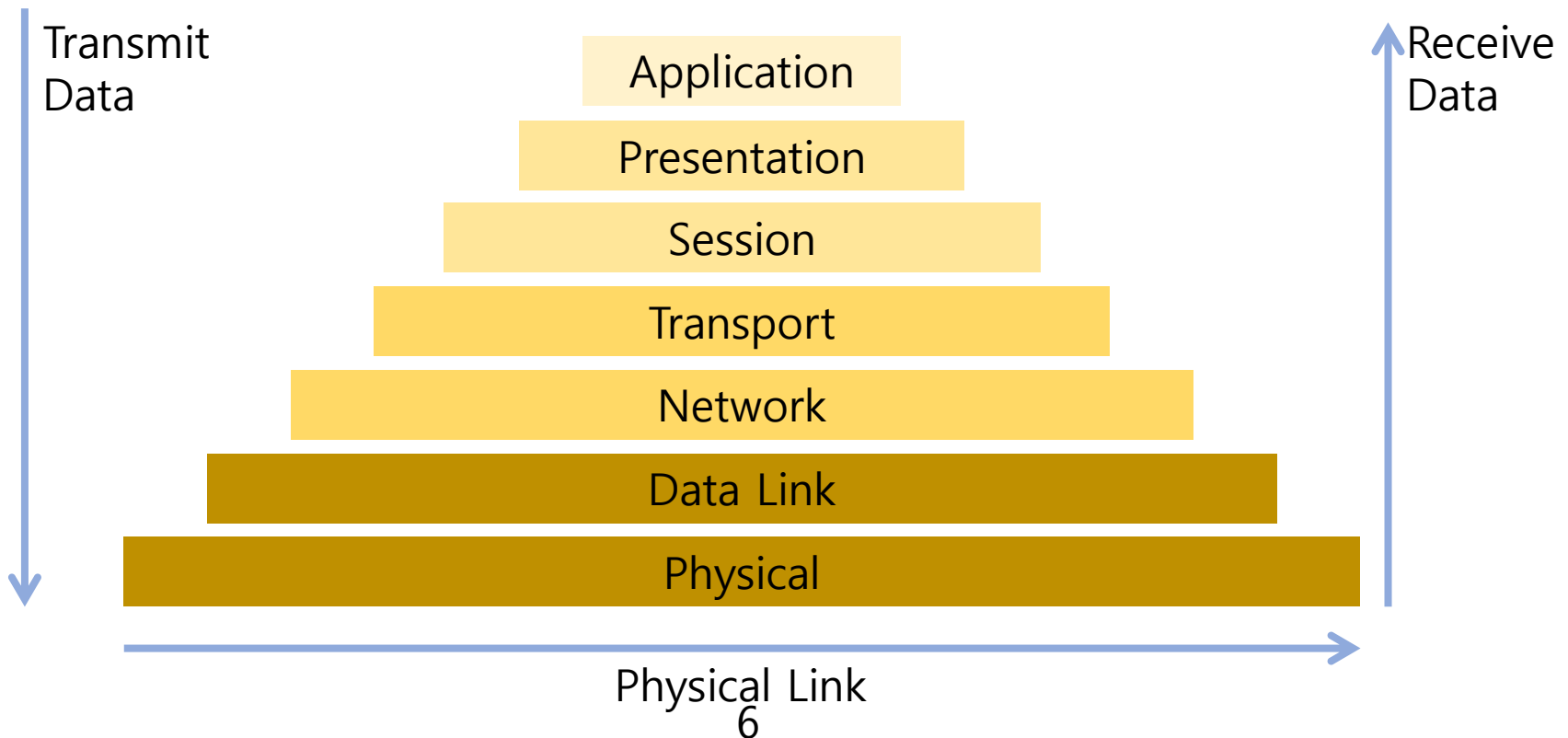
- 계층도



## 1.2. 계층 구조

### • OSI 7 계층

- ✓ 국제표준화기구(ISO)에서 개발한 계층
- ✓ Computer Network Protocol Design과 통신을 계층으로 나누어 설명
- ✓ 일반적으로 OSI 7 계층 모형이라 칭함



- **OSI 7 계층**

- ✓ 이 모델은 Protocol을 기능별로 나눈 것
- ✓ 각 계층은 하위 계층의 기능만을 이용하고, 상위 계층에게 기능을 제공
- ✓ 'Protocol stack' 혹은 'stack'은 이러한 계층들로 구성되는 Protocol System이 구현된 시스템
  - stack은 Hardware나 Software 혹은 둘의 혼합으로 구현 가능
  - 일반적으로 하위 계층들은 Hardware로, 상위 계층들은 Software로 구현



- OSI 7 계층

- ✓ Physical Layer

- Network의 기본 Network Hardware 전송 기술
- Network의 높은 수준의 기능의 논리 데이터 구조를 기초로 하는 필수 계층
- 다양한 특징의 Hardware 기술이 접목되어 있기에 OSI Architecture에서 가장 복잡한 계층으로 간주

- ✓ Data Link Layer

- Point to Point 간의 reliability 전송을 보장하기 위한 계층으로 CRC 기반의 error control과 flow control이 필요
- Network 위의 개체들 간 데이터를 전달하고, Physical Layer에서 발생할 수 있는 오류를 찾아 내고, 수정하는 데 필요한 기능적, 절차적 수단을 제공
- error control / retransmission / flow control

## 1.2. 계층 구조

### ✓ Network Layer

- 여러개의 Node를 거칠 때마다 경로를 찾아주는 역할을 하는 계층
- 다양한 길이의 데이터를 Network들을 통해 전달하고, 그 과정에서 Transport layer가 요구하는 Quality Of Service를 제공하기 위한 기능적, 절차적 수단을 제공
- 주소부여(IP)
- 경로설정(Route)

### ✓ Transport Layer

- End to end의 사용자들이 reliability 데이터를 주고 받을 수 있도록 해 주어, 상위 계층들이 데이터 전달의 유효성이나 효율성을 생각하지 않도록 지원
- 패킷들의 전송이 유효한지 확인하고 전송 실패한 패킷들을 다시 전송
- 오류검출 및 복구와 흐름제어, 중복검사 등을 수행
- 패킷 생성

## 1.2. 계층 구조

- ✓ Session Layer
- ✓ Presentation Layer
- ✓ Application Layer
  - Application Process와 직접 관계하여 일반적인 Application Service를 수행
  - 일반적인 Application Service는 관련된 Application Process사이의 전환을 제공
  - Network Software UI 부분
  - User의 입출력(I/O)부분

### • OSEK COM 계층 구조

#### ✓ IL ( Interaction Layer )

- 메시지 전송을 위한 API를 제공
- 특정 응용 프로그램 데이터를 담은 메시지를 통해 통신
- 메시지와 메시지의 특성은 OSEK 구현 언어 (OIL)를 통해 모두 정적으로 구성
- 내부 통신은 모두 IL에 의해 처리
- 외부 통신은 IL이 하나 이상의 메시지를 I-PDU에 포장해서 underlying layer에 전달
- 내부-외부 통신의 경우는 같은 메시지를 외부와 내부로 동시에 보낼 때를 뜻함
- 메시지 객체를 기반으로 모든 메시지를 관리
- IL과 하위 계층간의 통신 데이터는 하나 이상의 메시지로 구성된 I-PDU로 구성
- 초기화,데이터 전송,통신 관리 등을 위한 API를 제공
- Callout을 통해 기능 확장 가능

## 1.2. 계층 구조

### ✓ Network Layer

- 사용되는 통신 Protocol에 의존적
- 메시지 segmentation/recombination 및 acknowledgement 등을 처리
- flow control mechanisms을 제공
- Data Link Layer에서 제공하는 Service를 이용
- IL의 모든 특징을 지원하기 위한 최소의 requirements만 정의

### ✓ Data Link Layer

- Network를 통해 승인되지 않은 각각의 Data frame을 전송하기 위한 Service를 상위 계층에 제공
- Network Management를 위한 service를 제공
- IL의 모든 특징을 지원하기 위한 최소의 requirements만 정의

## 1. OSEK/VDX COM

1.1. 개요

1.2. 계층 구조

1.3. Interaction Layer

1.4. Message Reception

1.5. Message Transmission

1.6. Communication System Management

1.7 Deadline Monitoring

## 2. OSEK/VDX NM

2.1. 개요

2.2. Direct OSEK NM

2.3. Indirect OSEK NM

## 1.3. Interaction Layer

- 개요

- ✓ Message는 OIL을 통해 static 정의
- ✓ IL은 수신자가 Message Data를 즉시 사용 할 수 있도록 제공
- ✓ external communication 경우
  - 하나 이상의 할당된 Message를 Interaction Layer Protocol Data Units (I-PDU)에 압축하여 underlying layer에 전달
- ✓ internal communication 경우
  - external communication 기능의 Sub-set

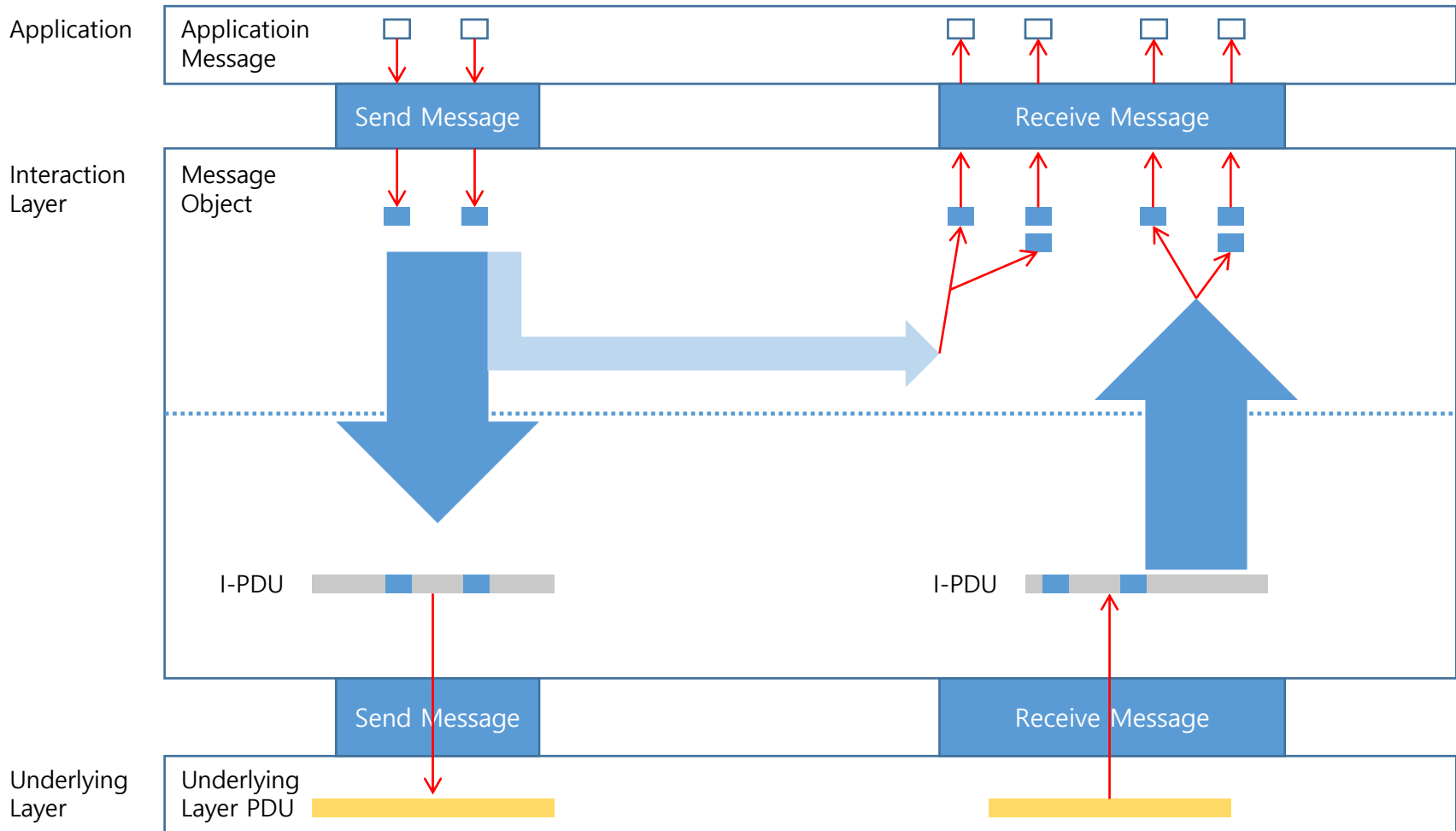
- **Communication concept**

- ✓ 메시지의 전송과 수신은 각각 OSEK OS의 Task와 ISR
- ✓ 발신 메시지 객체에서 발신되고 수신 메시지 객체로부터 수신
- ✓ 다수 대 다수의 통신을 지원
- ✓ Zero-length 메시지도 허용
- ✓ 메시지는 세 가지 종류가 존재
- ✓ 수신 메시지 객체는 두 가지 메시지 종류가 존재
- ✓ 외부 메시지 송수신을 monitoring하기 위한 Deadline Monitoring 메커니즘 제공
- ✓ 몇 개의 filter 알고리즘 제공



# 1.3. Interaction Layer

- Communication concept



- **Message 종류**
  - ✓ Static-length message
    - 지정된 크기로 가장 많이 쓰는 메시지
  - ✓ Zero-length message
    - 메시지 값이 없고 단지 신호를 알리기 위한 메시지
  - ✓ Dynamic-length message
    - 메시지의 길이가 가변적이거나 최대 길이제한을 가진 메시지

- **Configuration**

- ✓ 메시지, 발신자, 수신자의 구성은 시스템 생성 시 정의
- ✓ 실행 중에 메시지 추가 또는 삭제 불가능
- ✓ I-PDU에 적재된 메시지의 변경 불가능
- ✓ 3가지 구성요소
  - 메시지의 전달 특성 구성 및 I-PDU의 전송모드
  - PDU로의 메시지 적재(packing)
  - 수신자에 의한 queue 사용 및 queue의 크기
- ✓ 단일 CPU의 구성은 OIL에서 서술

## 1. OSEK/VDX COM

### 1.1. 개요

### 1.2. 계층 구조

### 1.3. Interaction Layer

### 1.4. Message Reception

### 1.5. Message Transmission

### 1.6. Communication System Management

### 1.7 Deadline Monitoring

## 2. OSEK/VDX NM

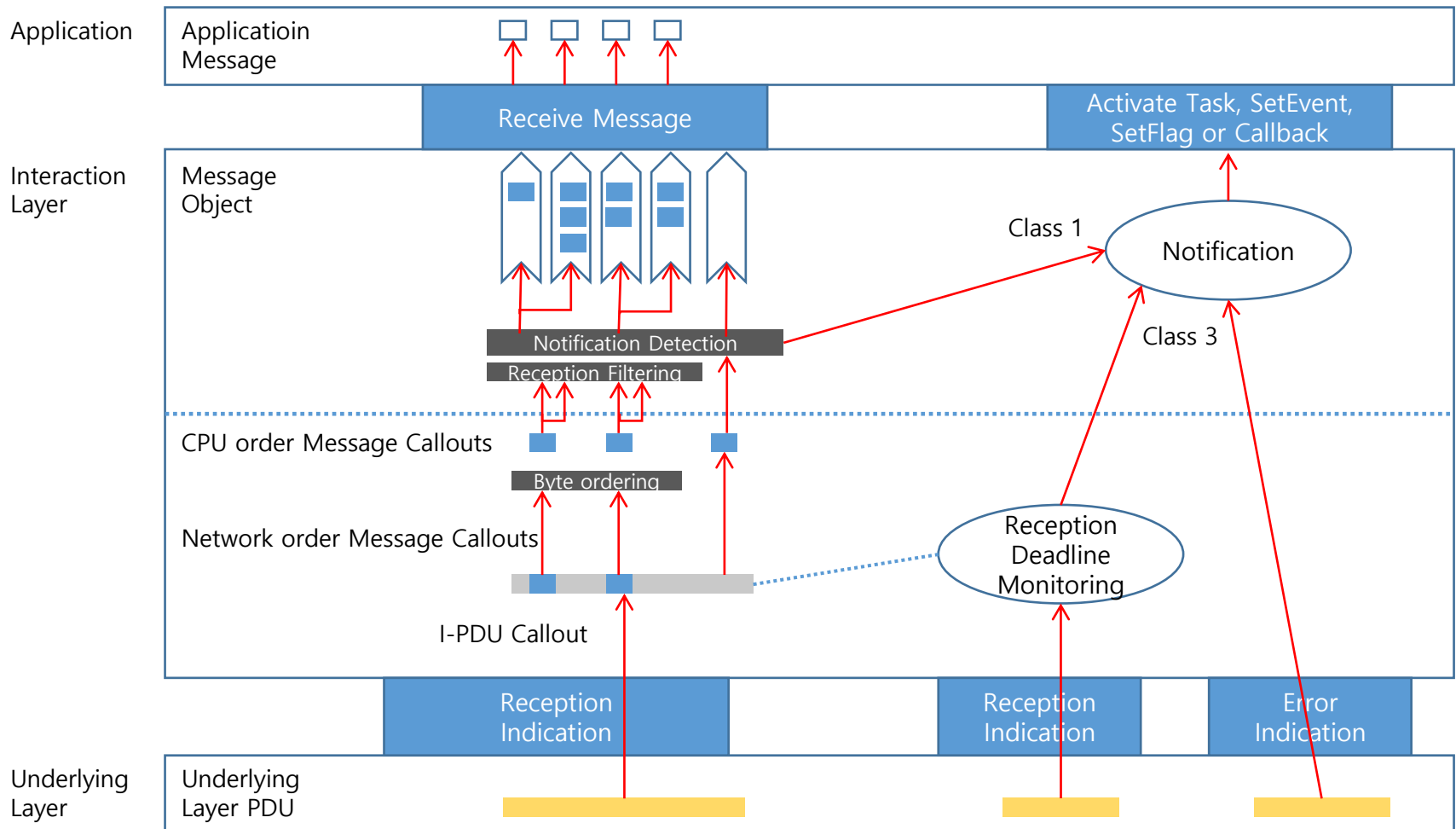
### 2.1. 개요

### 2.2. Direct OSEK NM

### 2.3. Indirect OSEK NM

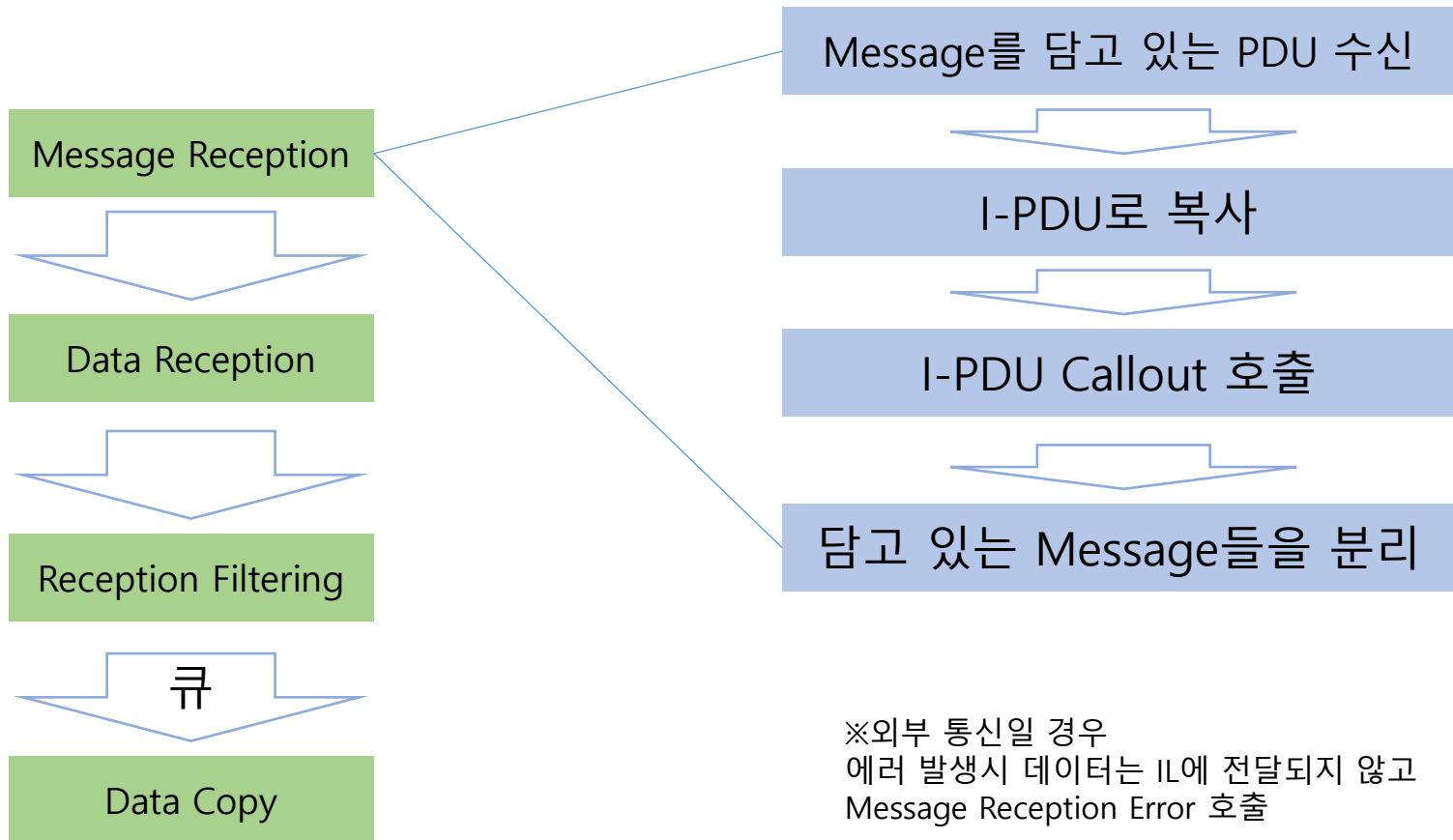
# 1.4. Message Reception

## • Message Reception 구성도



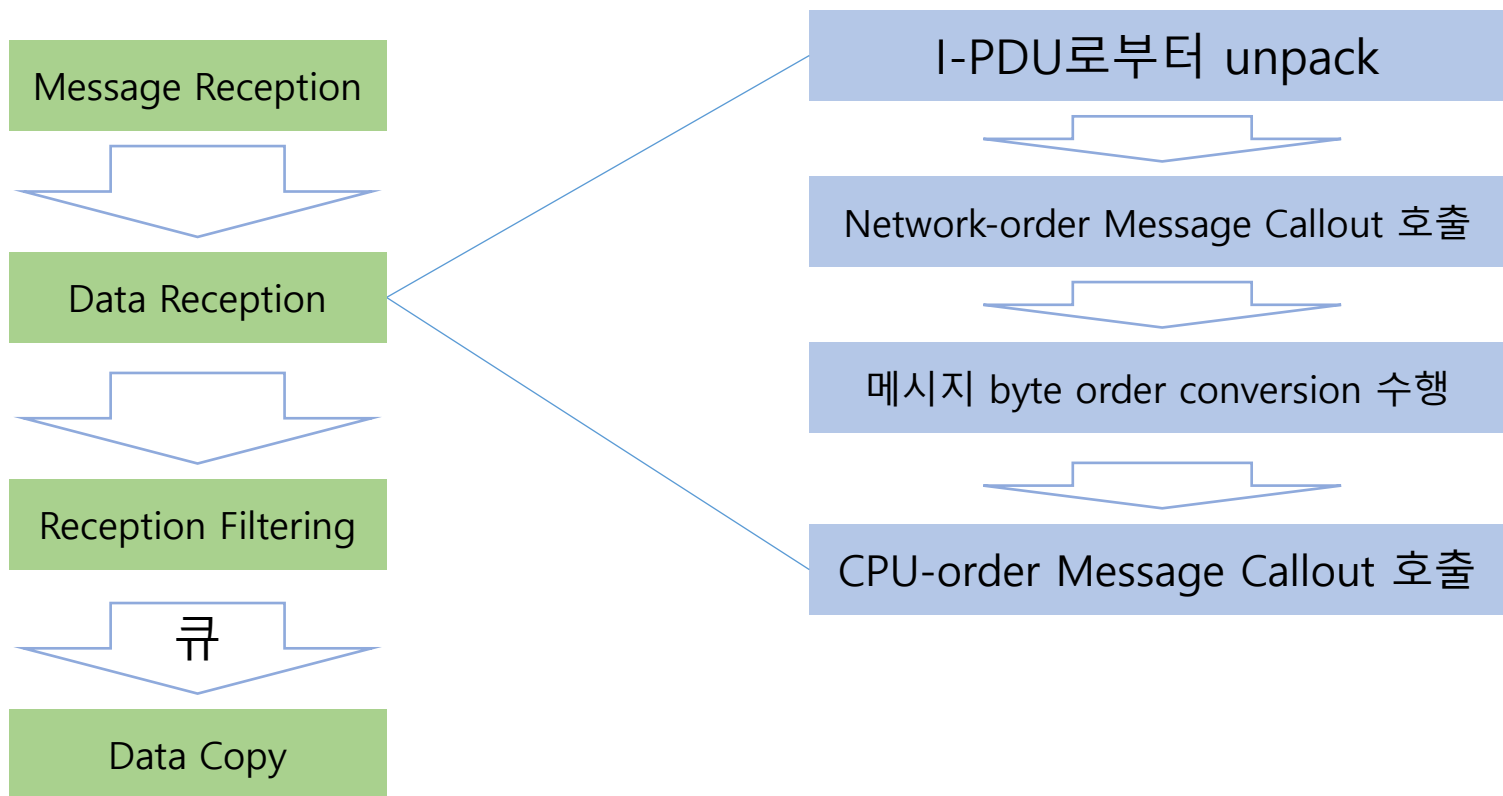
## 1.4. Message Reception

### • Message Reception 흐름도



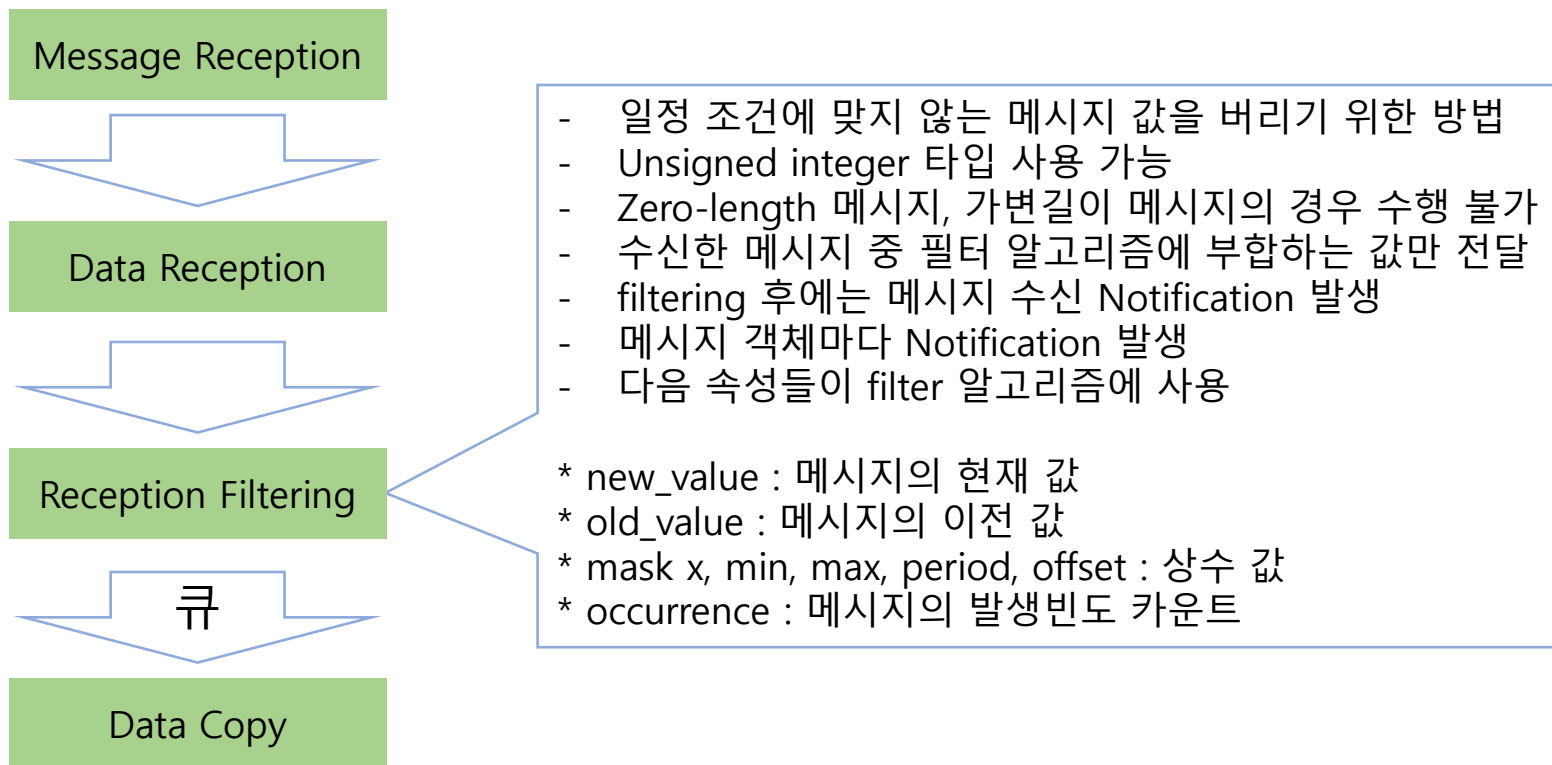
## 1.4. Message Reception

### • Message Reception 흐름도



## 1.4. Message Reception

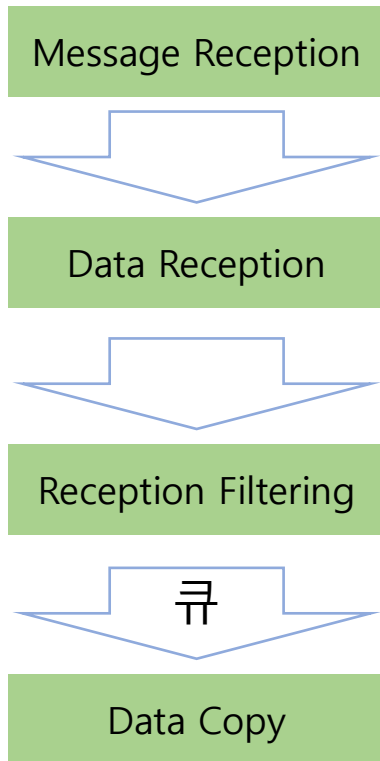
### • Message Reception 흐름도





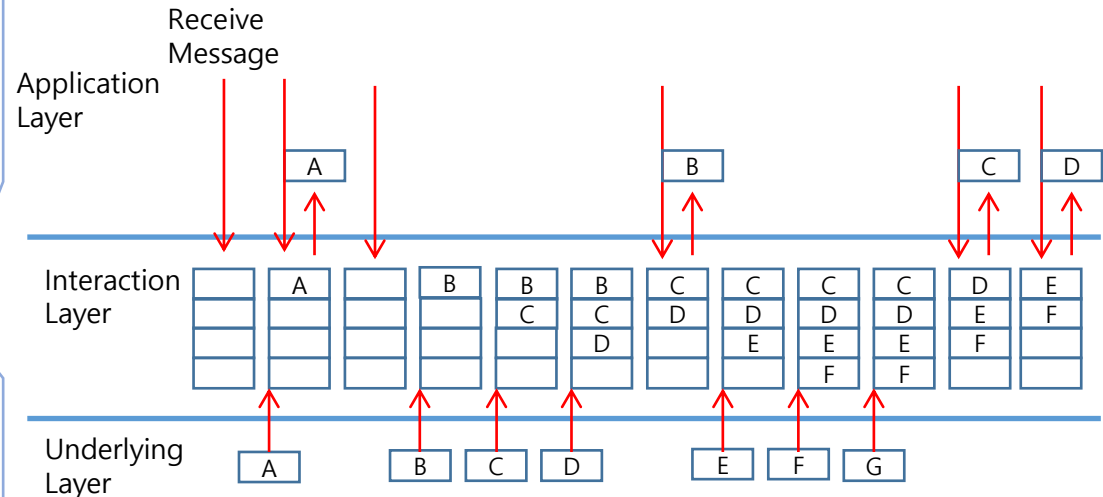
# 1.4. Message Reception

## • Message Reception 흐름도



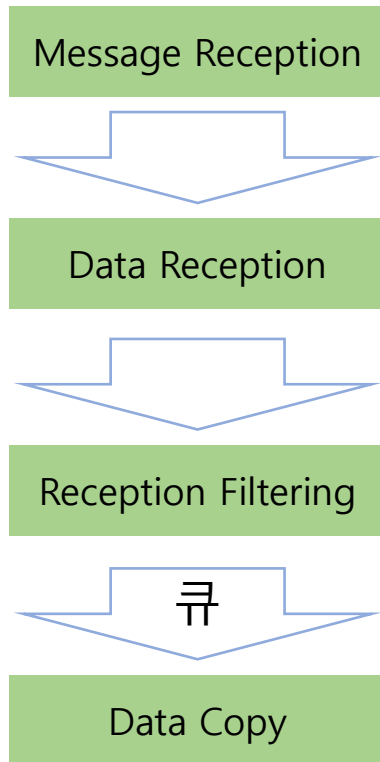
### 1. Queued 메시지

- FIFO 큐처럼 동작
- 큐가 꽉 차지 않은 상태라면 순차적으로 저장
- 큐가 꽉 차면 새로 도착한 메시지는 잃어버리게 되고 ReceiveMessage가 호출될 때 잃어버린 것을 알림



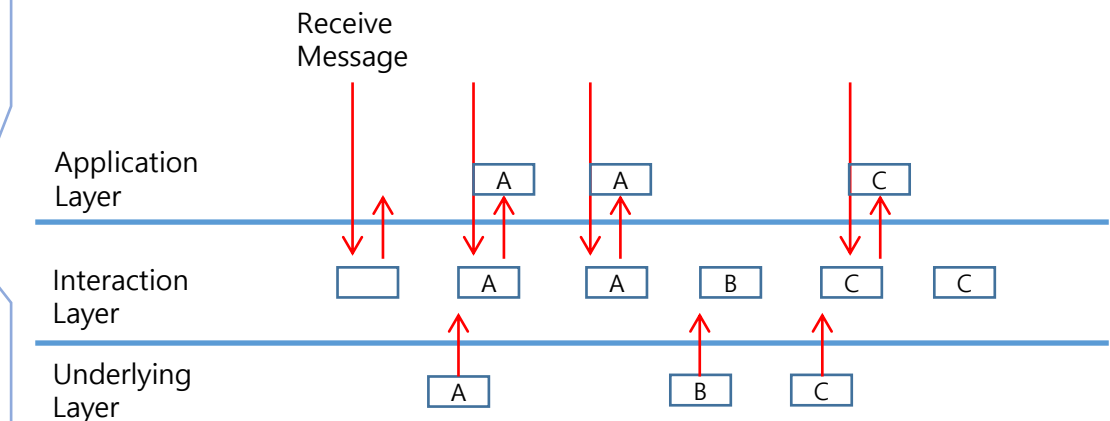
# 1.4. Message Reception

## • Message Reception 흐름도



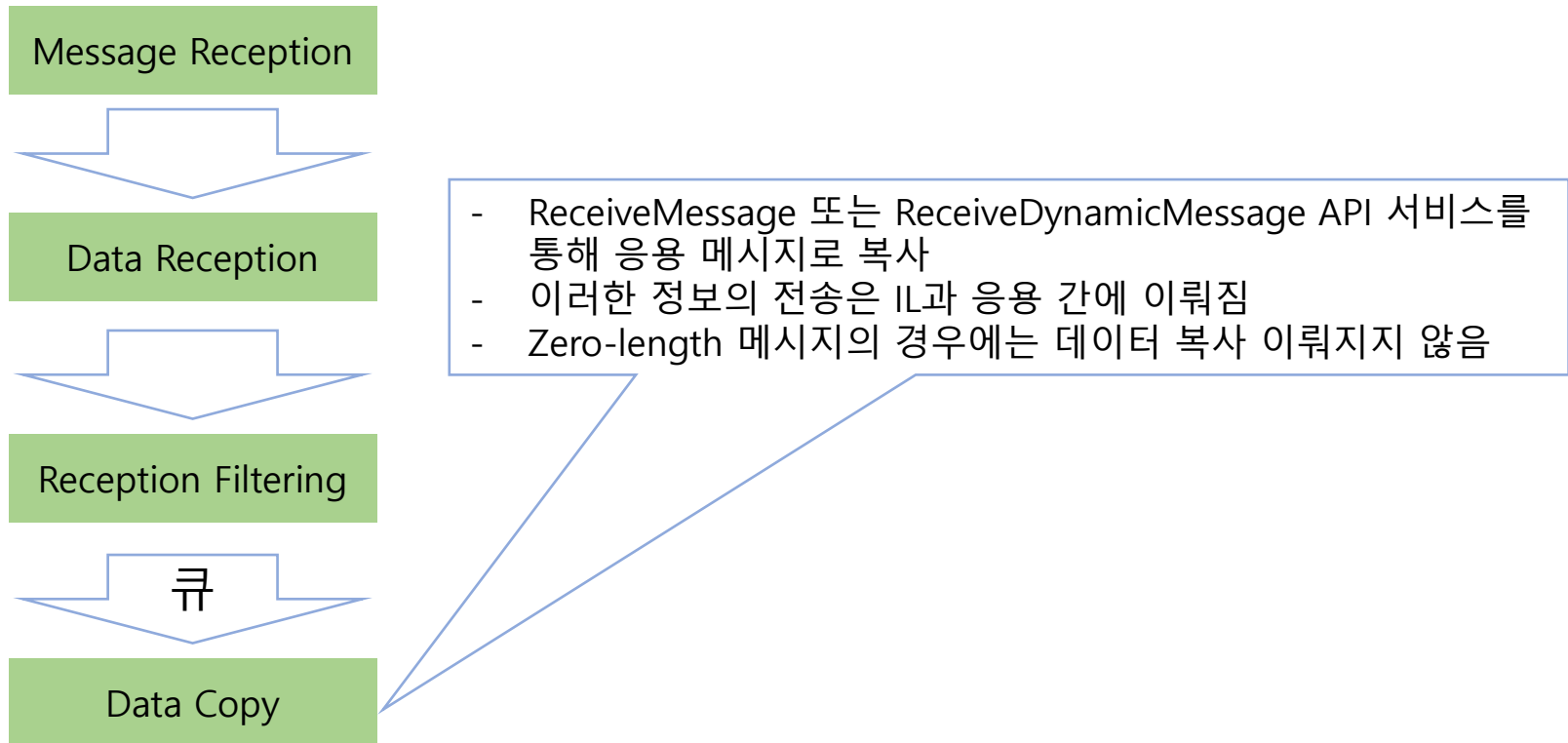
### 1. UnQueued 메시지

- 메시지가 수신되면 저장을 하고 수시로 읽기 가능
- 새로운 메시지 수신되면 기존 메시지 위에 덮어씀



## 1.4. Message Reception

### • Message Reception 흐름도



# 1.4. Message Reception

- Message filter

Algorithm Reference	Algorithm	Description
F_Always	True	NO Filtering is performed so that the message always passes
F_Never	False	The filter removes all messages
F_MaskedNewEqualsX	$(\text{new\_value} \& \text{mask}) == x$	Pass messages whose masked value is equal to a specific value
F_MaskedNewDiffersX	$(\text{new\_value} \& \text{mask}) != x$	Pass messages whose masked value is not equal to a specific value
F_NewIsEqual	$\text{New\_value} == \text{old\_value}$	Pass messages which have not changed
F_NewIsDifferent	$\text{New\_value} != \text{old\_value}$	Pass messages which have changed
F_MaskedNewEqualsMaskedOld	$(\text{new\_value} \& \text{mask}) == (\text{old\_value} \& \text{mask})$	Pass message where the masked value has not changed
F_MaskedNewDiffersMaskedOld	$(\text{new\_value} \& \text{mask}) != (\text{old\_value} \& \text{mask})$	Pass message where the masked value has changed
F_NewIsWithin	$\text{Min} \leq \text{new\_value} \leq \text{max}$	Pass a message if its value is within a predefined boundary
F_NewIsOutside	$(\text{min} > \text{new\_value}) \text{ OR } (\text{new\_value} > \text{max})$	Pass a message if its value is outside a predefined boundary
F_NewIsGreater	$\text{New\_value} > \text{old\_value}$	Pass a message if its value has increased
F_NewIsLessOrEqual	$\text{New\_value} \leq \text{old\_value}$	Pass a message if its value has not increased
F_NewIsLess	$\text{New\_value} < \text{old\_value}$	Pass a message if its value has decreased
F_NewIsGreaterOrEqual	$\text{New\_value} \geq \text{old\_value}$	Pass a message if its value has not decreased
F_OneEveryN	$\text{Occurrence \% period} == \text{offset}$	

## 1.4. Message Reception

- **Notification**

- ✓ 송신부와 수신부에 있는 메시지 객체(message object)마다 설정
- ✓ 전송을 위해 메시지를 포함한 I-PDU를 감시하기 위해서 또는 수신을 위해 메시지 객체를 감시하기 위해 실행

- **Notification Class**

- ✓ Notification Class 1 – 메시지 수신( Message Reception )
- ✓ Notification Class 2 – 메시지 전송( Message Transmission )
- ✓ Notification Class 3 – 메시지 수신 에러 ( Message Reception Error )
- ✓ Notification Class 4 – 메시지 전송 에러 ( Message Transmission Error )

## 1.4. Message Reception

### • Notification mechanism

#### ✓ Notification mechanism은 네 가지를 지원

- Callback Routine : IL은 응용 프로그램에서 제공하는 Callback routine 호출
- Task 활성화 : IL은 응용 프로그램 Task를 활성화
- Event 설정 : IL은 응용프로그램 Task를 위한 이벤트를 설정
- Flag
  - 설정되면 COM\_TRUE, 그렇지 않으면 COM\_FALSE
  - 리턴 클래스 1과 3에서는 ReceiveMessage 및 ReceiveDynamicMessage, 클래스 2와 4에서는 SendMessage, SendDynamicMessage 및 SendZeroMessage를 호출하여 플래그를 리셋

#### ✓ Interface for callback routines

- Interrupt Level 또는 Task Level에서 실행
- Task 뿐만 아니라 Interrupt Service Routines을 위한 System 기능의 OS 제한 사항이 적용
- COMCallback(CallbackRoutineName) {}

## 1. OSEK/VDX COM

1.1. 개요

1.2. 계층 구조

1.3. Interaction Layer

1.4. Message Reception

1.5. Message Transmission

1.6. Communication System Management

1.7 Deadline Monitoring

## 2. OSEK/VDX NM

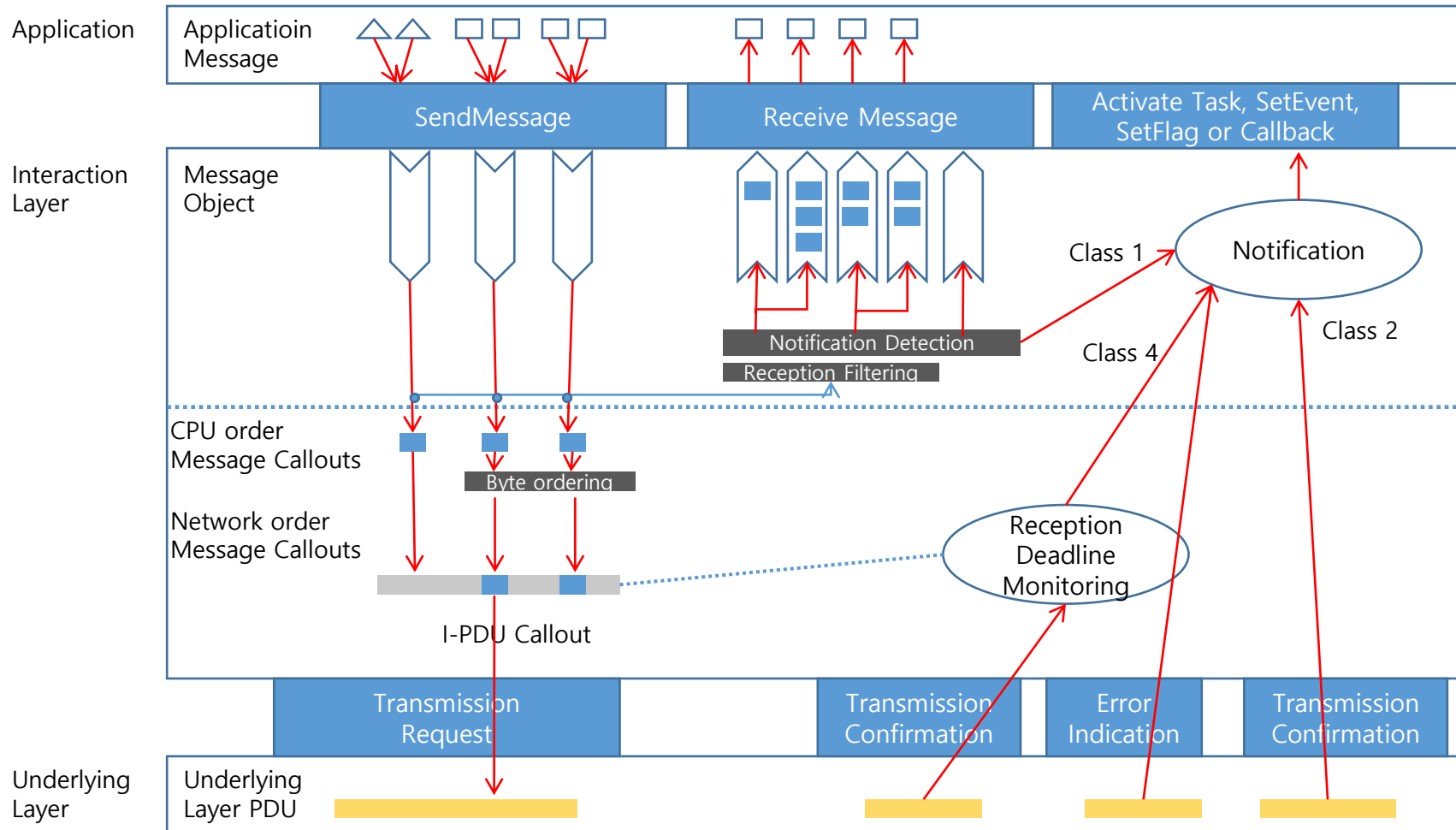
2.1. 개요

2.2. Direct OSEK NM

2.3. Indirect OSEK NM

# 1.5. Message Transmission

## • Message Transmission 구성도

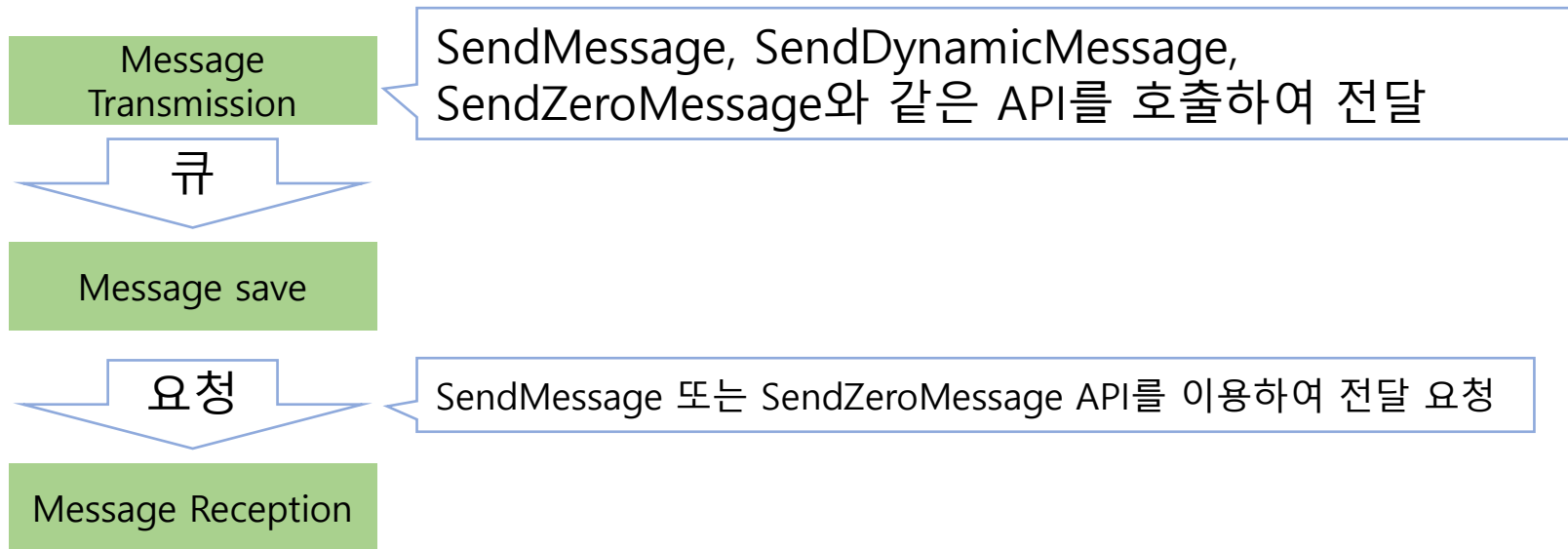




## 1.5. Message Transmission

### • Message Transmission 흐름도

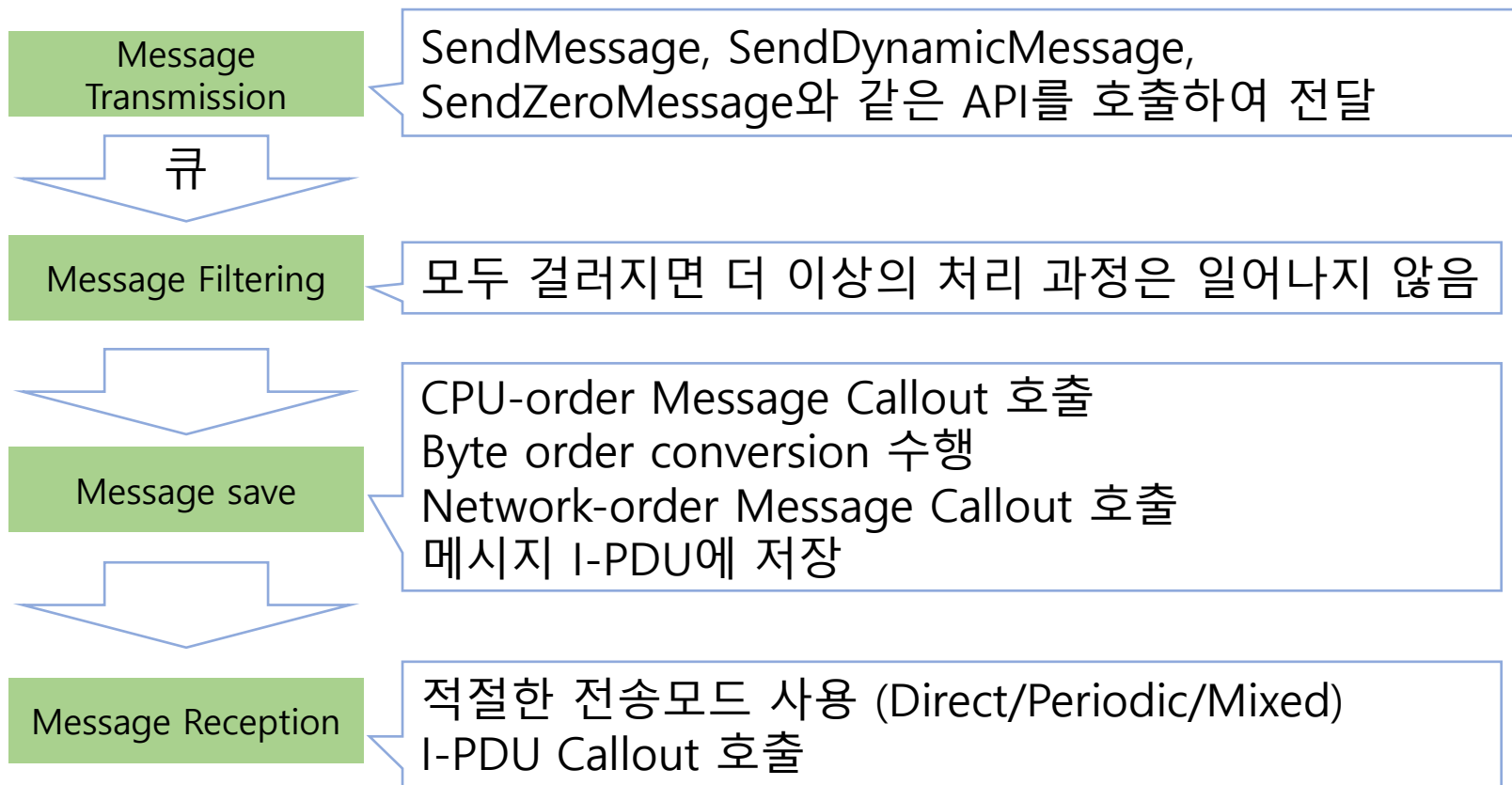
- ✓ Transfer of internal messages



## 1.5. Message Transmission

### • Message Transmission 흐름도

- ✓ Transfer properties for external communication

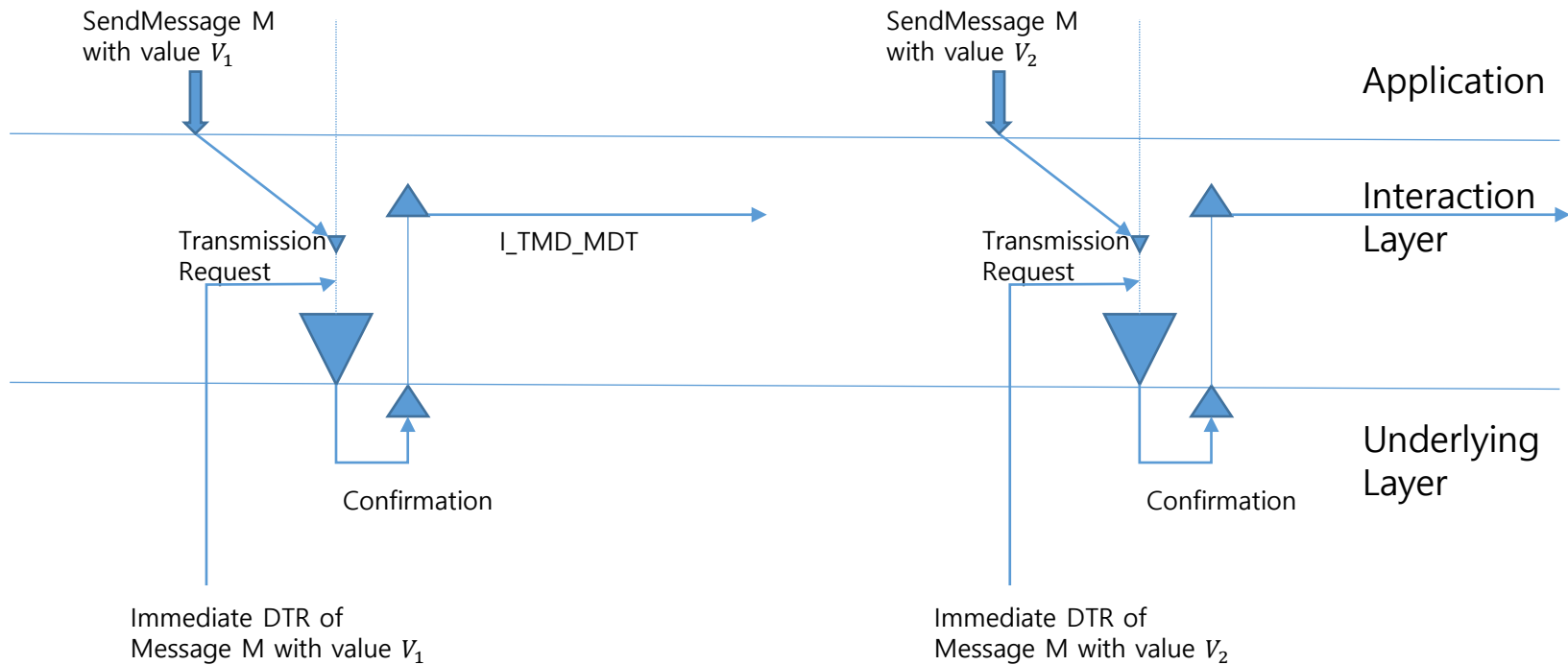


# 1.5. Message Transmission

## • Transmission Modes

### ✓ Direct Transmission Mode

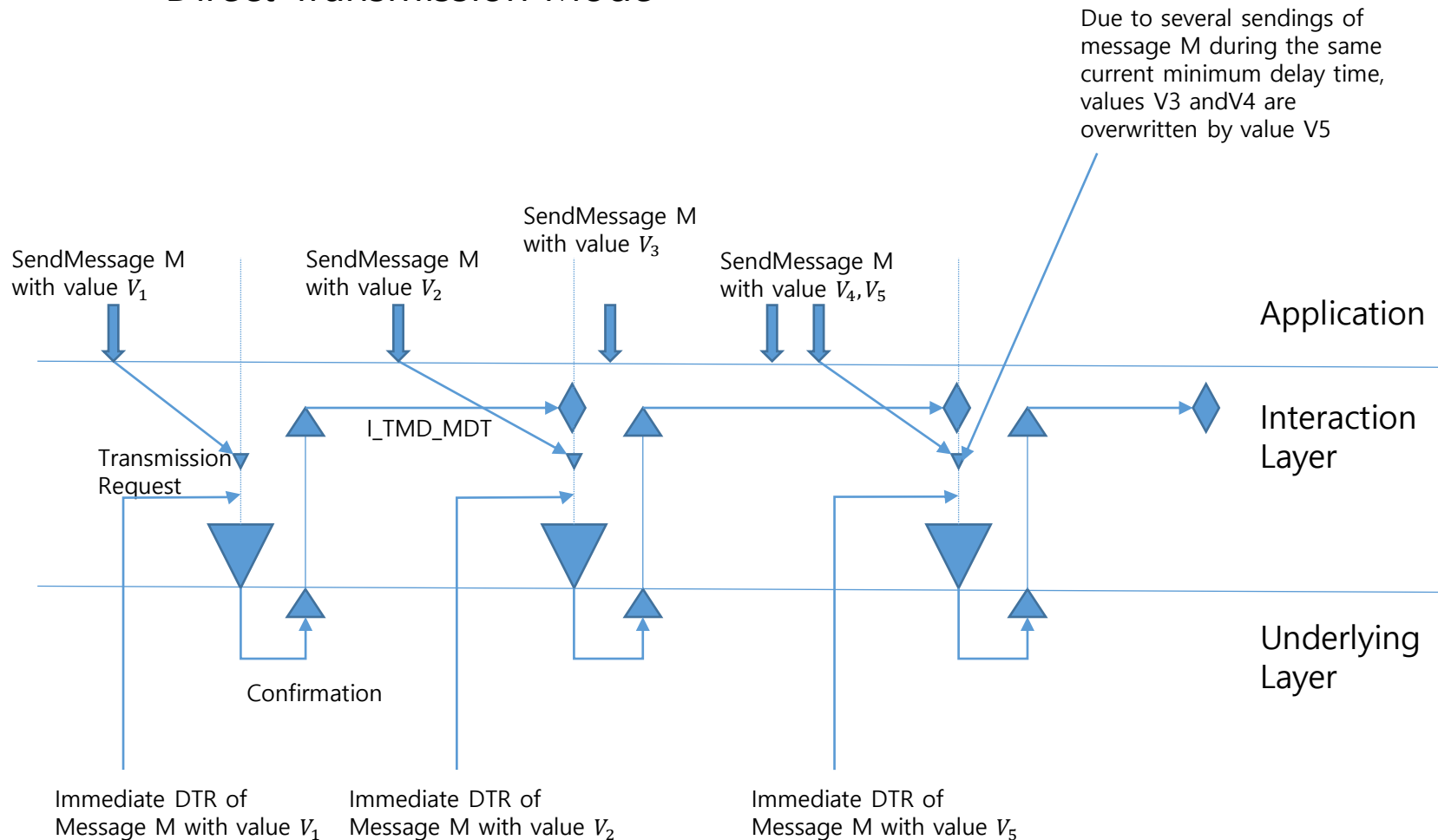
- \*DTR : Direct Transmission Request
- \*PTR : Period Transmission Request
- \*I\_TMD\_MDT : A minimum delay time between transmissions
- \*I\_TMP\_TPD : Periodic Transmission Mode Time Period
- \*I\_TMM\_TPD : Mixed Transmission Mode Time Period
- \*I\_TMP\_TOF : Periodic Transmission Mode Time Offset



## • Transmission Modes

### ✓ Direct Transmission Mode

- \*DTR : Direct Transmission Request
- \*PTR : Period Transmission Request
- \*I\_TMD\_MDT : A minimum delay time between transmissions
- \*I\_TMP\_TPD : Periodic Transmission Mode Time Period
- \*I\_TMM\_TPD : Mixed Transmission Mode Time Period
- \*I\_TMP\_TOF : Periodic Transmission Mode Time Offset

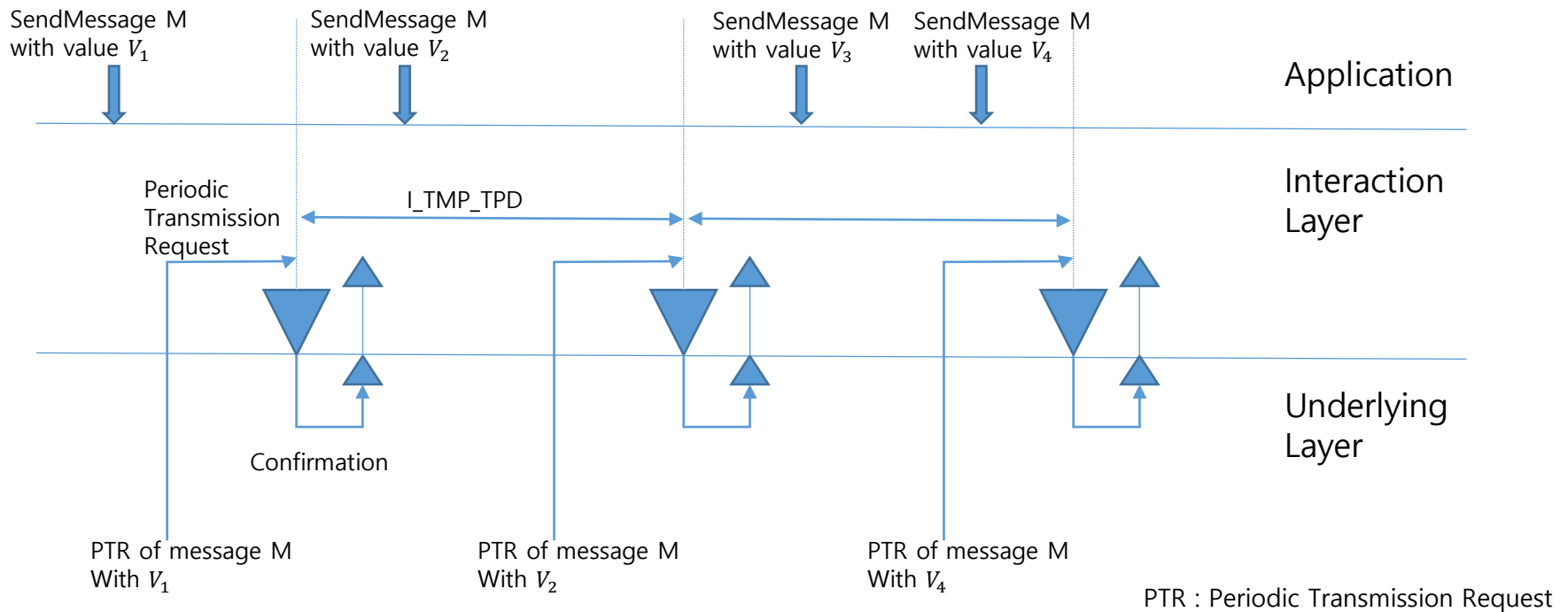


# 1.5. Message Transmission

## • Transmission Modes

### ✓ Periodic Transmission Mode

- \*DTR : Direct Transmission Request
- \*PTR : Periodic Transmission Request
- \*I\_TMD\_MDT : A minimum delay time between transmissions
- \*I\_TMP\_TPD : Periodic Transmission Mode Time Period
- \*I\_TMM\_TPD : Mixed Transmission Mode Time Period
- \*I\_TMP\_TOF : Periodic Transmission Mode Time Offset

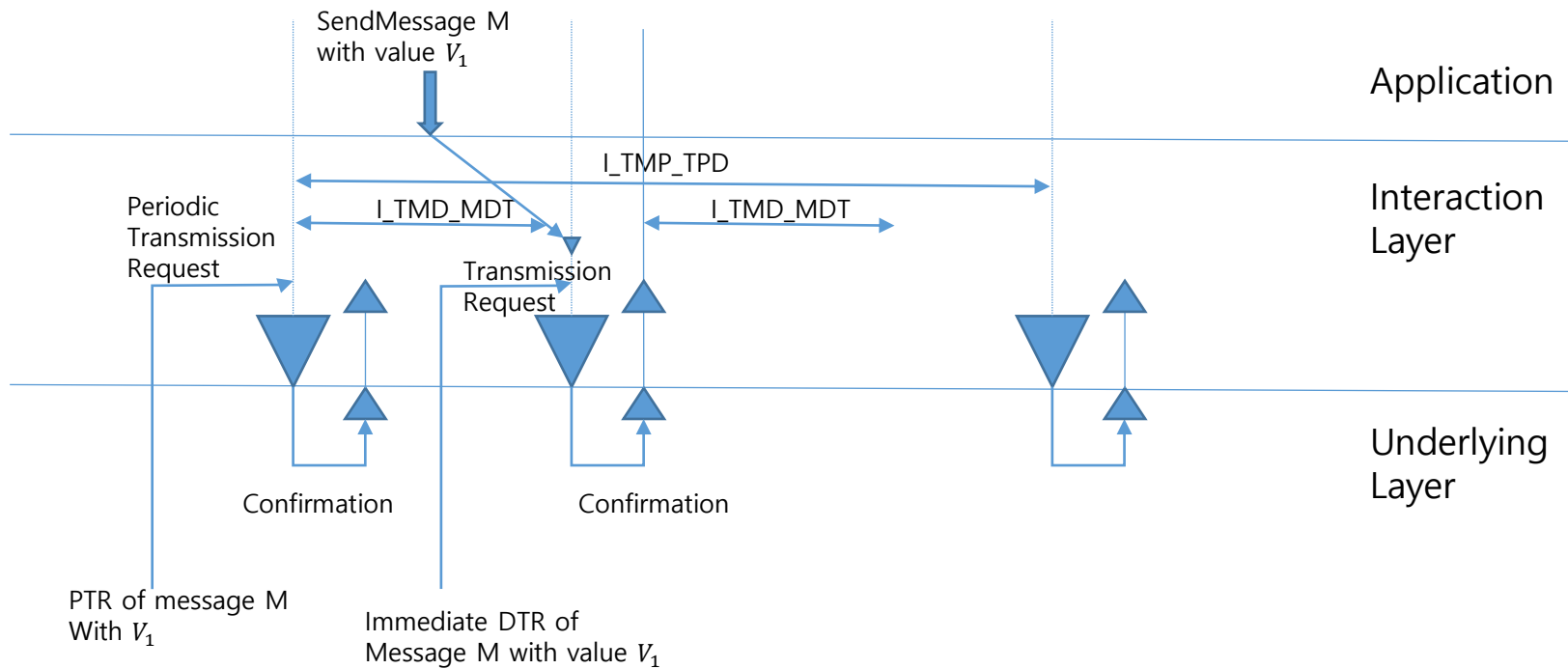


# 1.5. Message Transmission

## • Transmission Modes

### ✓ Mixed Transmission Mode

- \*DTR : Direct Transmission Request
- \*PTR : Periodic Transmission Request
- \*I\_TMD\_MDT : A minimum delay time between transmissions
- \*I\_TMP\_TPD : Periodic Transmission Mode Time Period
- \*I\_TMM\_TPD : Mixed Transmission Mode Time Period
- \*I\_TMP\_TOF : Periodic Transmission Mode Time Offset



## 1. OSEK/VDX COM

### 1.1. 개요

### 1.2. 계층 구조

### 1.3. Interaction Layer

### 1.4. Message Reception

### 1.5. Message Transmission

### 1.6. Communication System Management

### 1.7 Deadline Monitoring

## 2. OSEK/VDX NM

### 2.1. 개요

### 2.2. Direct OSEK NM

### 2.3. Indirect OSEK NM

- **Initialization / Shutdown**

- ✓ start up and shut down을 위한 4가지 service 제공

- StartCOM

- 내부의 OSEK COM 데이터 영역을 초기화 하고, 메시지 초기화 루틴을 호출하고 OSEK COM 모듈을 시작

- StopCOM

- OSEK COM을 종료시키고 사용하고 있던 자원을 놓음

- StartPeriodic 및 StopPriodic

- 주기적인 전송을 시작/종료

- InitMessage

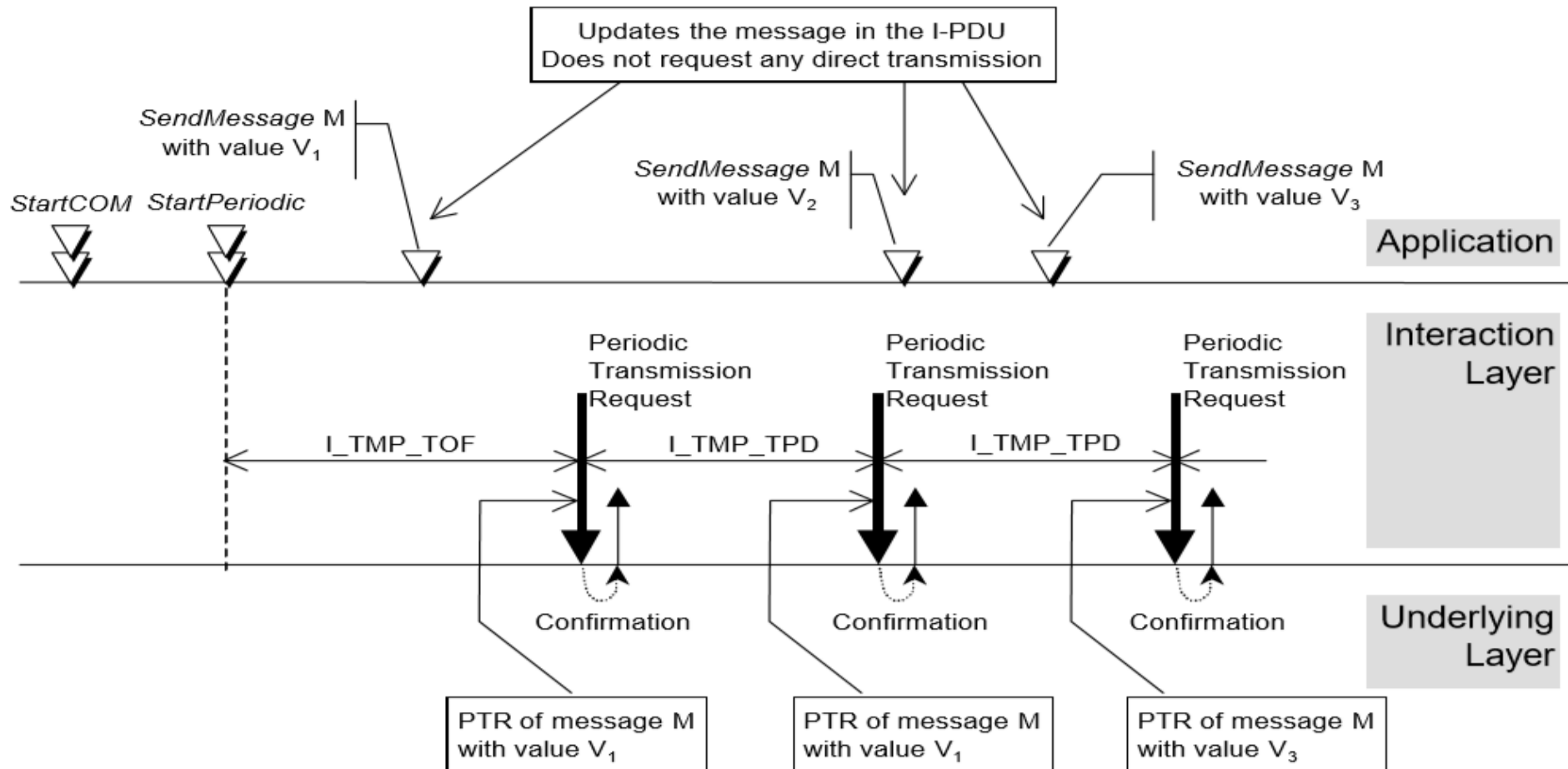
- 임의의 값으로 메시지를 초기화



# 1.6. Communication System Management

## • Periodic Transmission

- \*DTR : Direct Transmission Request
- \*PTR : Period Transmission Request
- \*I\_TMD\_MDT : A minimum delay time between transmissions
- \*I\_TMP\_TPD : Periodic Transmission Mode Time Period
- \*I\_TMM\_TPD : Mixed Transmission Mode Time Period
- \*I\_TMP\_TOF : Periodic Transmission Mode Time Offset



- **Error Handling**

- ✓ 일시적 또는 영구적인 error를 처리하기 위해 제공
- ✓ basic framework는 사전에 정의되어 있고 User가 완료
- ✓ User는 efficient centralised or decentralised error handling 선택 가능
- ✓ 두가지 error 종류
  - Application errors
    - IL이 요청 된 service를 올바르게 실행할 수 없지만 내부 데이터 가정 가능
    - centralised error treatment 호출
    - decentralised error treatment으로 status information 반환
  - Fatal errors
    - 더 이상 내부 데이터의 정확성을 가정 불가
    - centralised system shutdown 호출

- **Error Handling**

- ✓ 두가지 error checking 제공

- Standard error checking

- production phase에서 완전히 디버깅 된 application system에 사용

- Extended error checking

- development phase에서 불완전히 디버깅된 applications에 사용
- plausibility checks를 항상 시키지만 Standard error checking 보다 많은 memory와 실행 시간을 필요

- ✓ Error hook routine

- System service가 E\_OK가 아닌 StatusType 값을 반환 시 호출
- 사용자 정의 기능으로 사용자가 구현
- 구현에 따라 IL에 의해 호출 됨
- 필수이지만 OIL을 통해 구성 가능

## 1. OSEK/VDX COM

### 1.1. 개요

### 1.2. 계층 구조

### 1.3. Interaction Layer

### 1.4. Message Reception

### 1.5. Message Transmission

### 1.6. Communication System Management

### 1.7. Deadline Monitoring

## 2. OSEK/VDX NM

### 2.1. 개요

### 2.2. Direct OSEK NM

### 2.3. Indirect OSEK NM

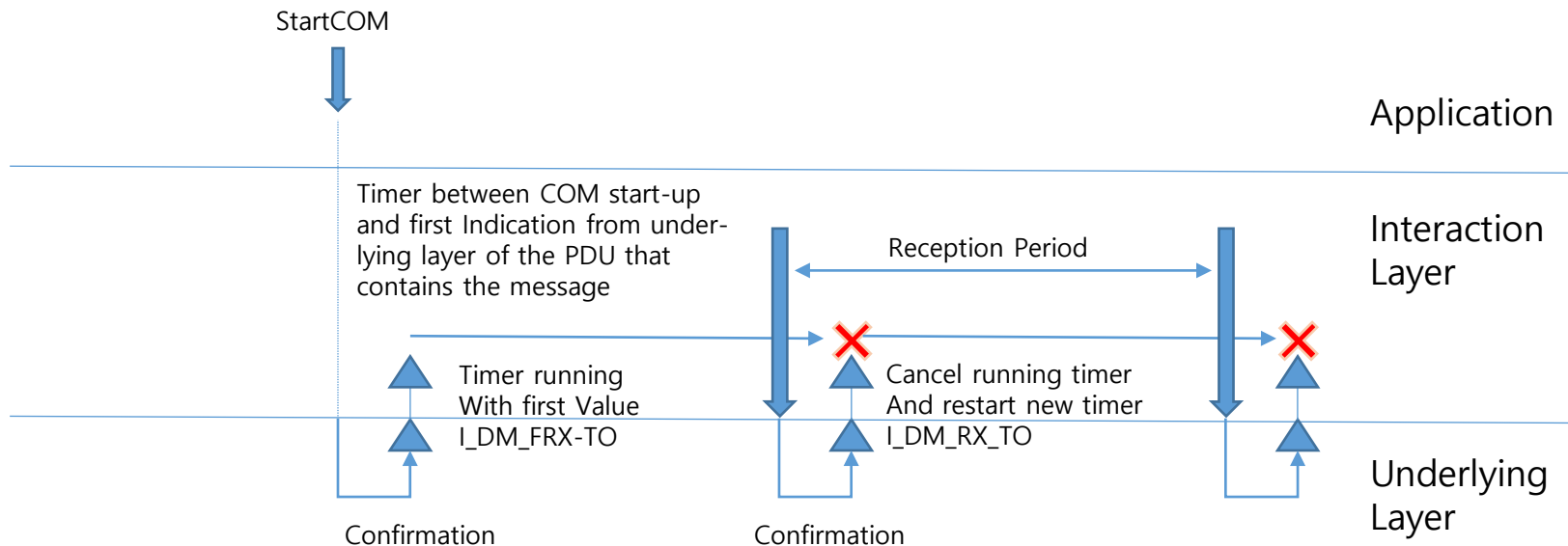
- **Reception Deadline Monitoring**

- ✓ time frame에 맞게 receive가 periodic messages 를 받았는지 검증
- ✓ Message마다 구성되며 Message가 포함된 I-PDU의 reception을 Monitoring
- ✓ Reception Deadline Monitoring은 external communication만 사용
- ✓ Reception이 없고 time-out 이 발생하면 Timer가 즉시 재시작
- ✓ 처음 monitor 된 시간 간격에 대한 Timer는 message object initialization tasks( StartCOM API )가 수행되면 시작

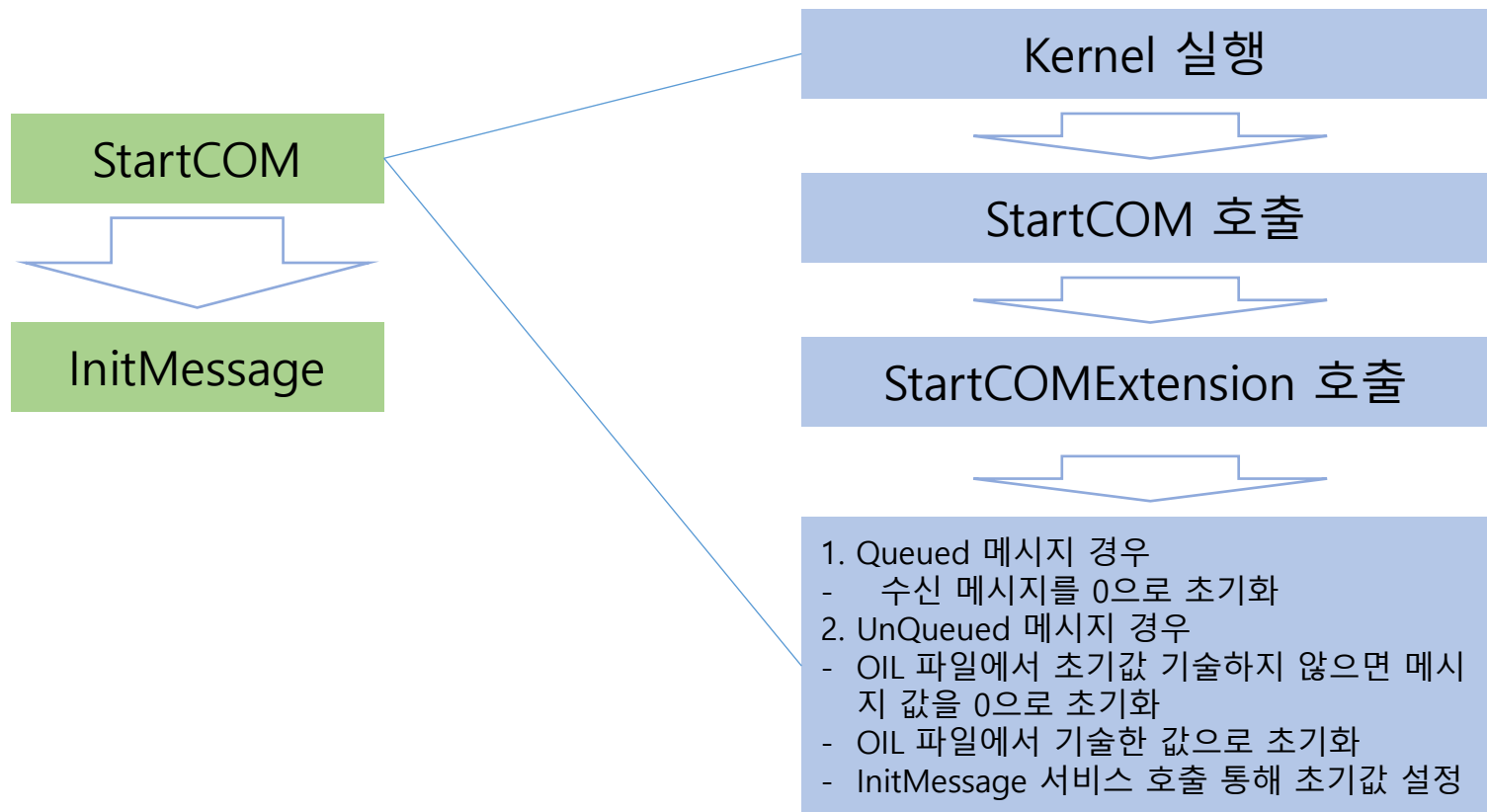
# 1.7 Deadline Monitoring

## • Reception Deadline Monitoring

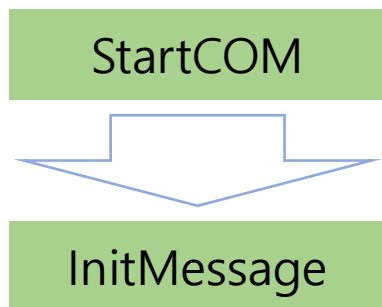
- ✓ Deadline Monitoring for periodic reception



## • Reception Deadline Monitoring



## • Reception Deadline Monitoring



1. OIL 파일에 의해 기술되는 크고 복잡한 초기값의 메시지를 초기화 하는데 사용
2. StartCOM 서비스가 호출된 후 또는 StopCOM이 호출되기 전 응용 프로그램 실행 중에 호출
3. 메시지 값을 0으로 초기화 하였거나 OIL 파일에서 특정 값으로 초기화한 후 재 초기화에 사용

1. 외부 전송 메시지일 경우
  - I-PDU 및 old\_value 메시지 필드는 특정 값으로 설정
2. 내부 전송 메시지일 경우
  - 초기화 진행 하지 않음
3. 수신 메시지일 경우
  - Unqueued 메시지의 메시지 객체는 특정 값으로 설정
  - 만일 old\_value를 사용한 필터 알고리즘이 unqueued 또는 queued 메시지에 대해 사용되면 old\_value는 특정 값으로 설정

※ 가변 길이 메시지 : 모든 메시지 초기화  
Queued 메시지 : 수신한 메시지를 0으로 설정



- **Transmission Deadline Monitoring**

- ✓ transmission requests (periodic or 기타) 가 time frame 안에 Network transmissions 수행 유무 검증
- ✓ IL은 I-PDU마다 monitoring 수행
- ✓ 따라서 Time-out은 I-PDU의 property
- ✓ 3가지 Transmission Deadline Monitoring
  - Direct Transmission Mode
  - Periodic Transmission Mode
  - Mixed Transmission Mode

## • Transmission Deadline Monitoring

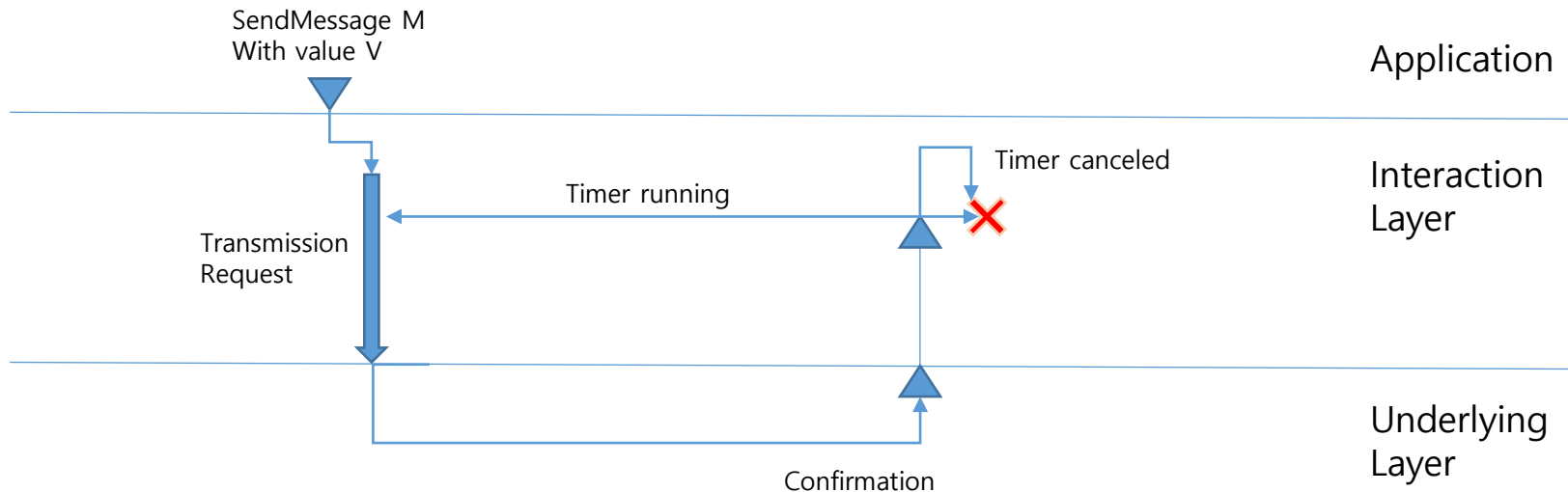
### ✓ Direct Transmission Mode

- SendMessage, SendDynamicMessage 또는 SendZeroMessage에 대한 각 호출을 monitoring
- 지정된 시간 간격 (I\_DM\_TMD\_TO)내에 underlying layer 에서 confirmation 발생하는지 확인
- monitoring timer는 SendMessage, SendDynamicMessage 또는 SendZeroMessage API Service 호출이 완료되면 시작
- Timer는 underlying layer에 의한 confirmation 전송 시에 취소
- Application는 적절한 notification mechanism으로 알림

## • Transmission Deadline Monitoring

### ✓ Direct Transmission Mode

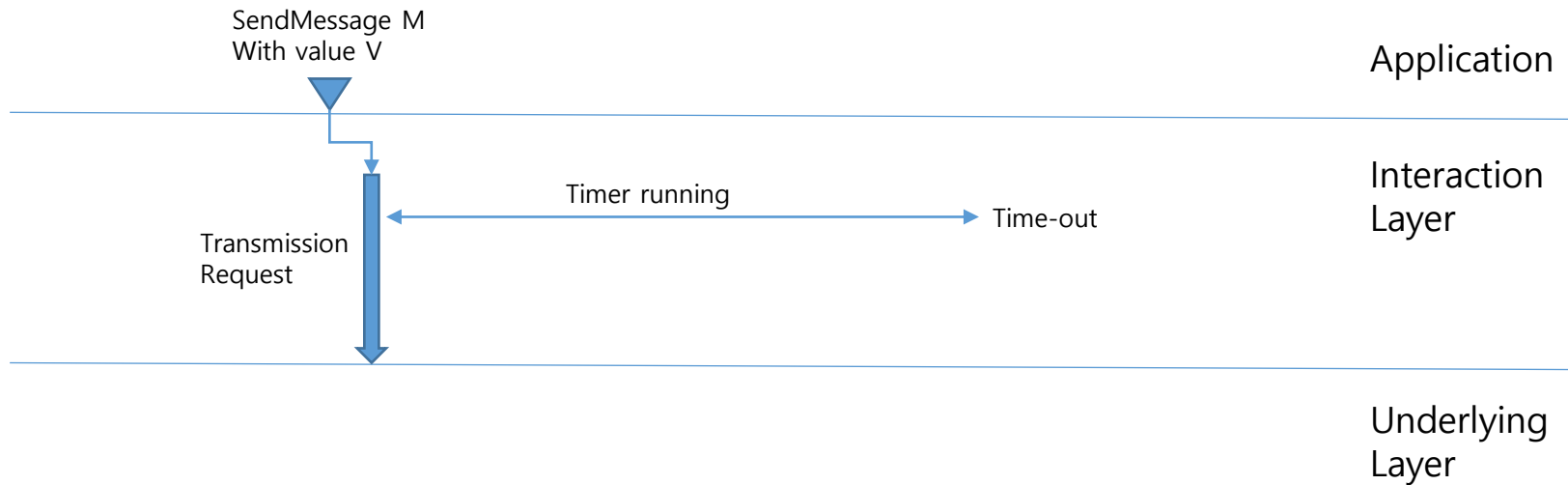
- example of a successful transmission



## • Transmission Deadline Monitoring

### ✓ Direct Transmission Mode

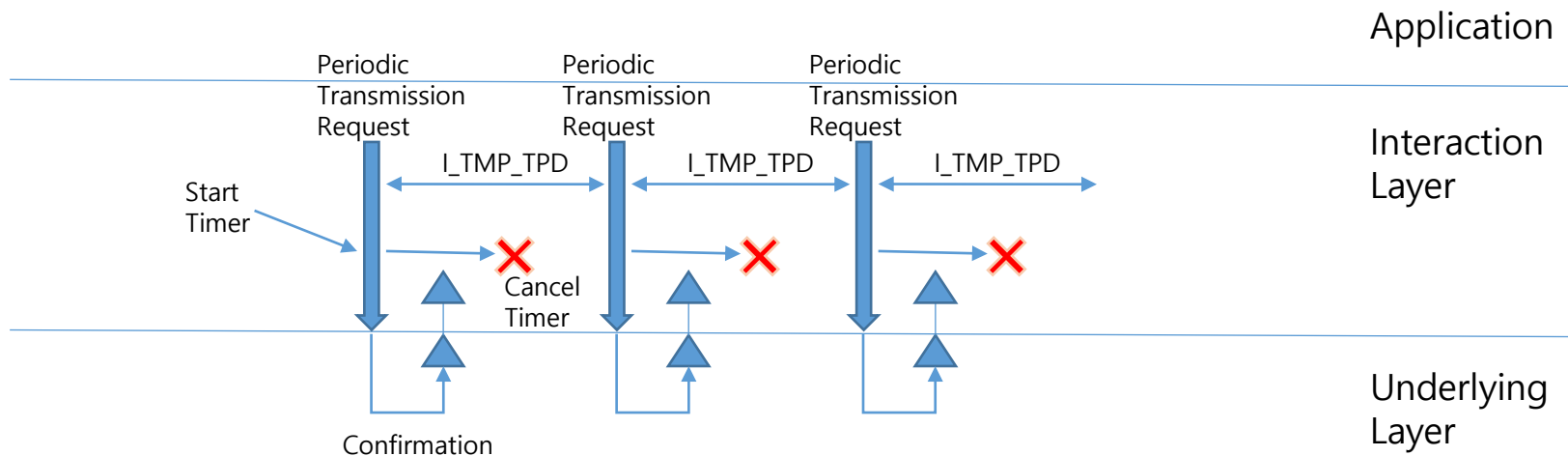
- example of a failed transmission



## • Transmission Deadline Monitoring

### ✓ Periodic Transmission Mode

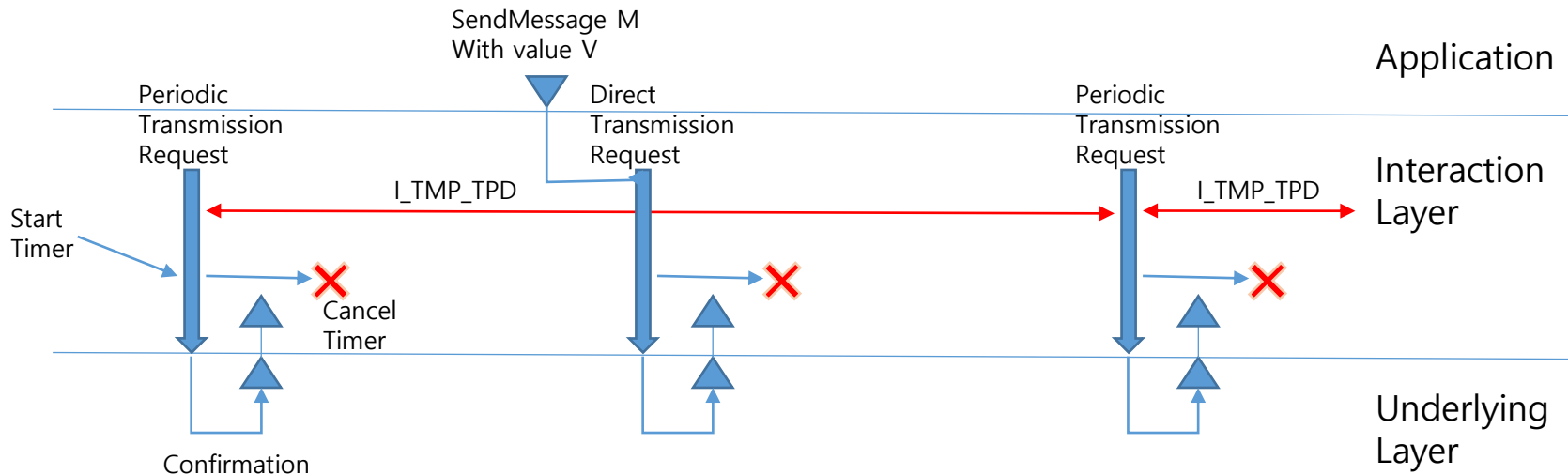
- successful transmission



## • Transmission Deadline Monitoring

### ✓ Mixed Transmission Mode

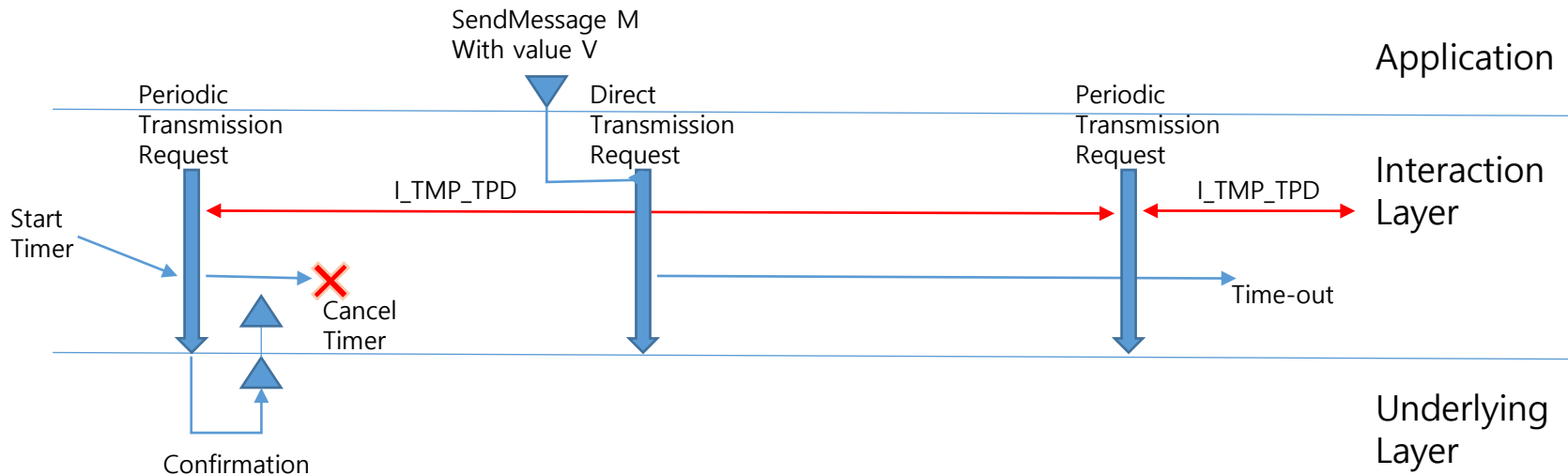
- successful transmissions



## • Transmission Deadline Monitoring

### ✓ Mixed Transmission Mode

- failed transmissions



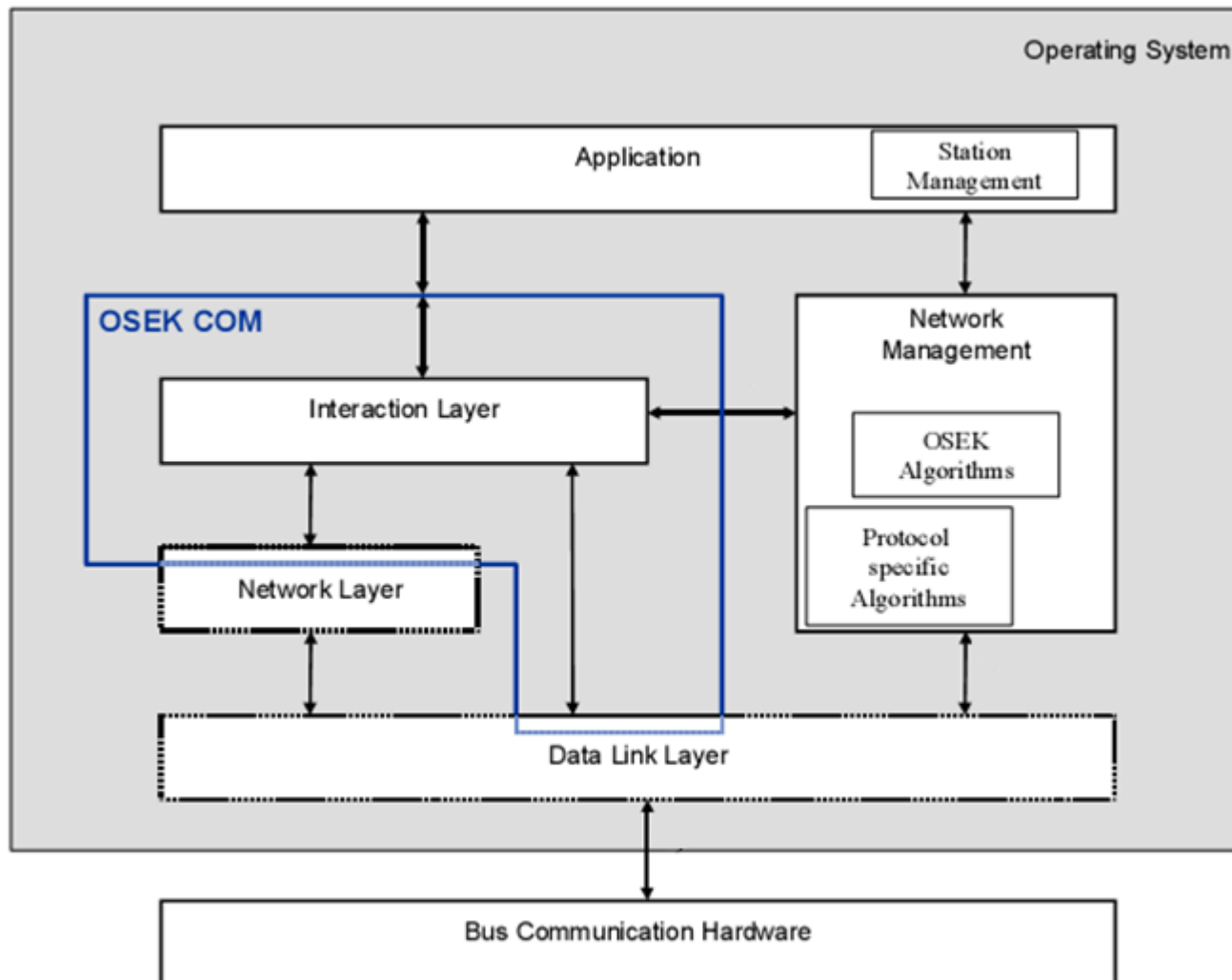
1. OSEK/VDX COM
  - 1.1. 개요
  - 1.2. 계층 구조
  - 1.3. Interaction Layer
  - 1.4. Message Reception
  - 1.5. Message Transmission
  - 1.6. Communication System Management
  - 1.7 Deadline Monitoring
2. OSEK/VDX NM
  - 2.1. 개요
  - 2.2. Direct OSEK NM
  - 2.3. Indirect OSEK NM



- **특징**

- ✓ 안전한 통신을 통해 기능을 보장하며, 배터리 방전을 줄이기 위해 다음과 같은 기능을 제공
  - Network의 Start-up 및 Shut-down의 동기화
  - ECU의 동작 모드의 관리 및 모니터링
  - ECU의 Sleep 및 Wakeup 제어를 통한 전원 관리
  - 두가지 모드 지원
  - Direct OSEK NM
  - Indirect OSEK NM

1. OSEK/VDX COM
  - 1.1. 개요
  - 1.2. 계층 구조
  - 1.3. Interaction Layer
  - 1.4. Message Reception
  - 1.5. Message Transmission
  - 1.6. Communication System Management
  - 1.7 Deadline Monitoring
2. OSEK/VDX NM
  - 2.1. 개요
  - 2.2. Direct OSEK NM
  - 2.3. Indirect OSEK NM



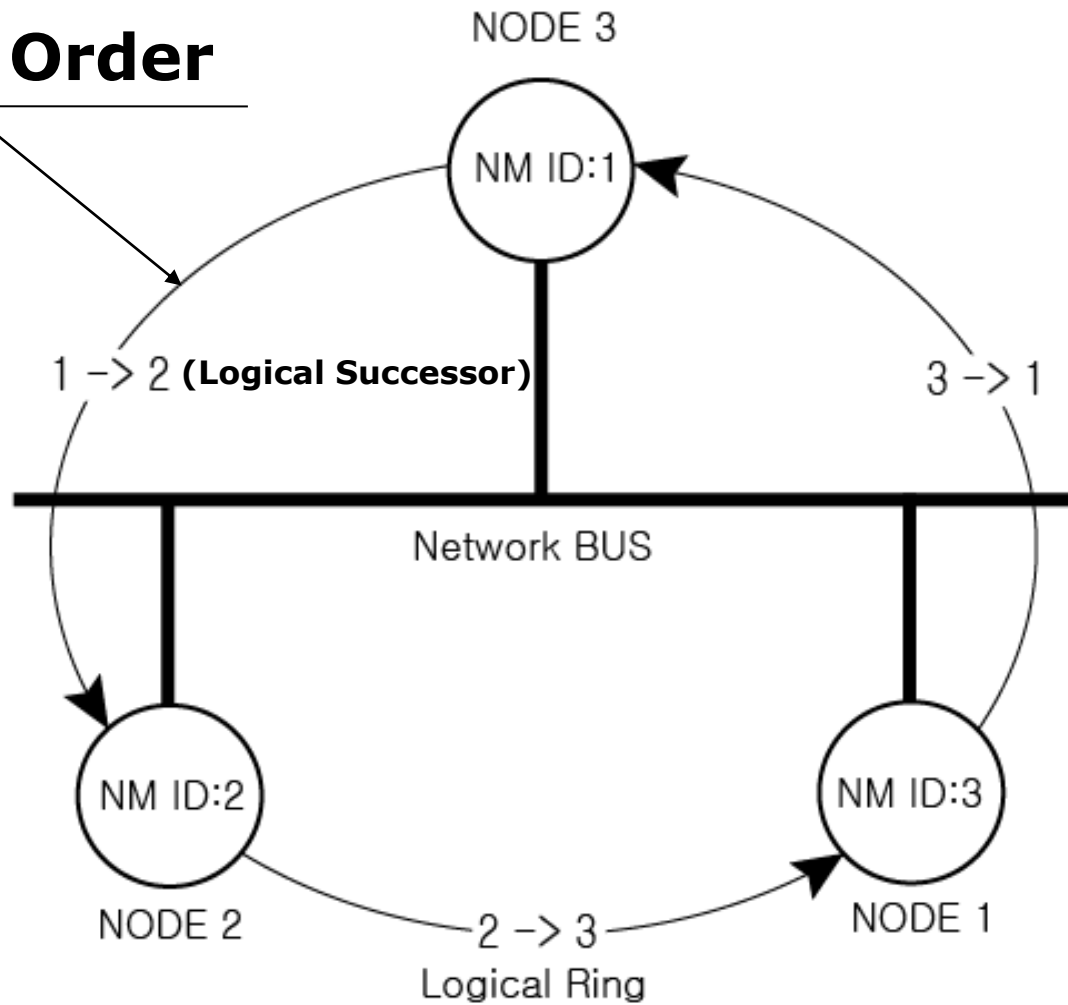
- Feature of direct NM

- ✓ Every node is actively monitored by every other node in the network
- ✓ The rate of the NM messages is controlled across the network in order to minimize bus load
- ✓ The nodes of direct NM transmit the NM message in periodically
- ✓ When a node receives the NM message, a node wait a time delay and then transmits its own NM message
- ✓ This mechanism is part of logic ring and make synchronize NM messages of network

- Logical ring

- ✓ In a logical ring the communication sequence is defined independently from the network structure
- ✓ Each node is determined its logical successor by numerical order then the logical sequence is formed
- ✓ The first node is the logical successor of the last node in the ring
- ✓ Any node must be able to send NM messages to all other nodes and receive messages from them

## Numerical Order



- **Alive message**

- ✓ An alive message introduces a new transmitter to the logical ring
- ✓ Receive alive message
  - Registration to the logical ring

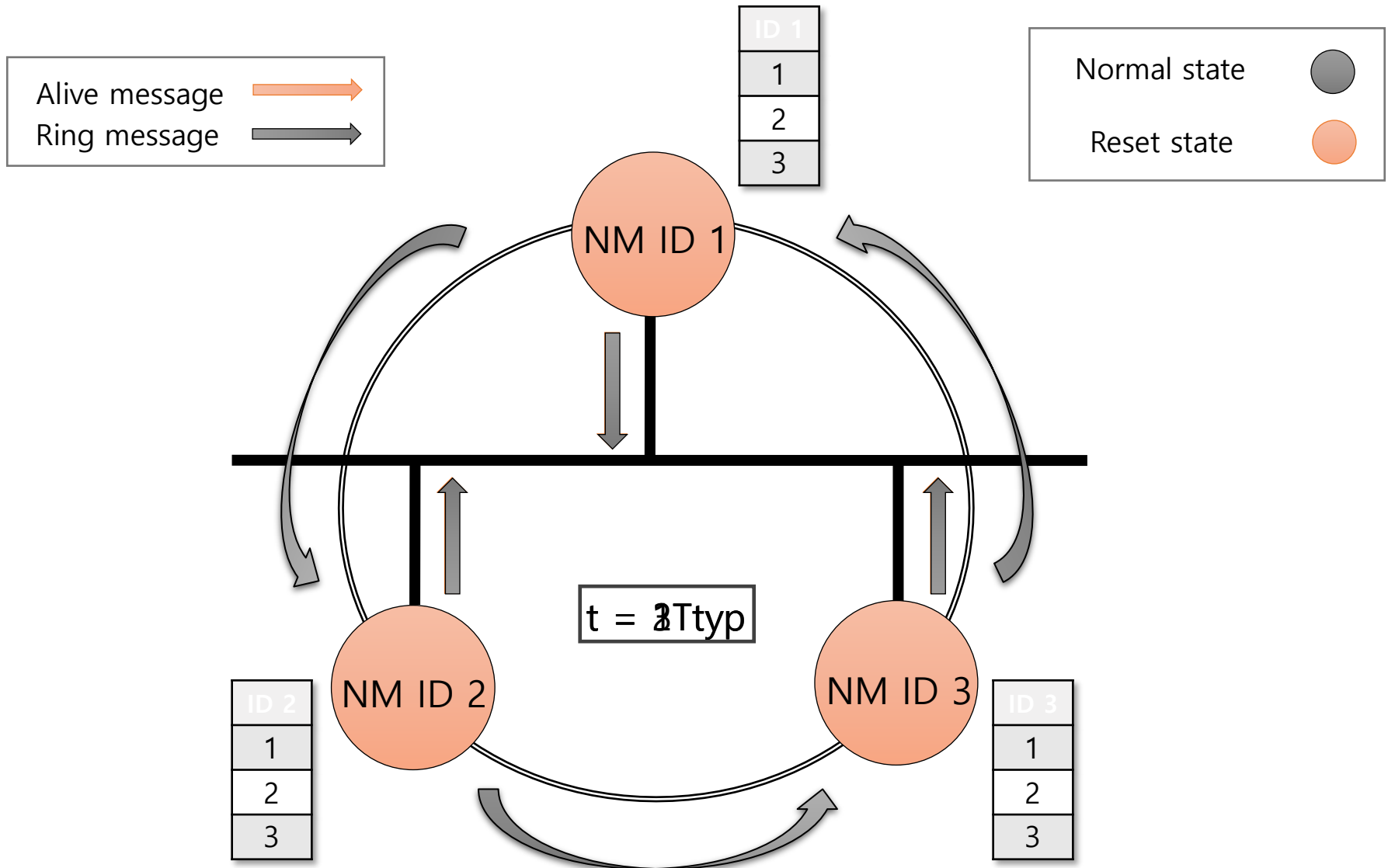
- **Ring message**

- ✓ When state is normal, node periodically transmit the ring message to logical successor
- ✓ Receive ring message
  - Interpretation as that the transmitter state is normal state
- ✓ Any nodes does not receive ring message
  - Interpretation as that the transmitter state is break down

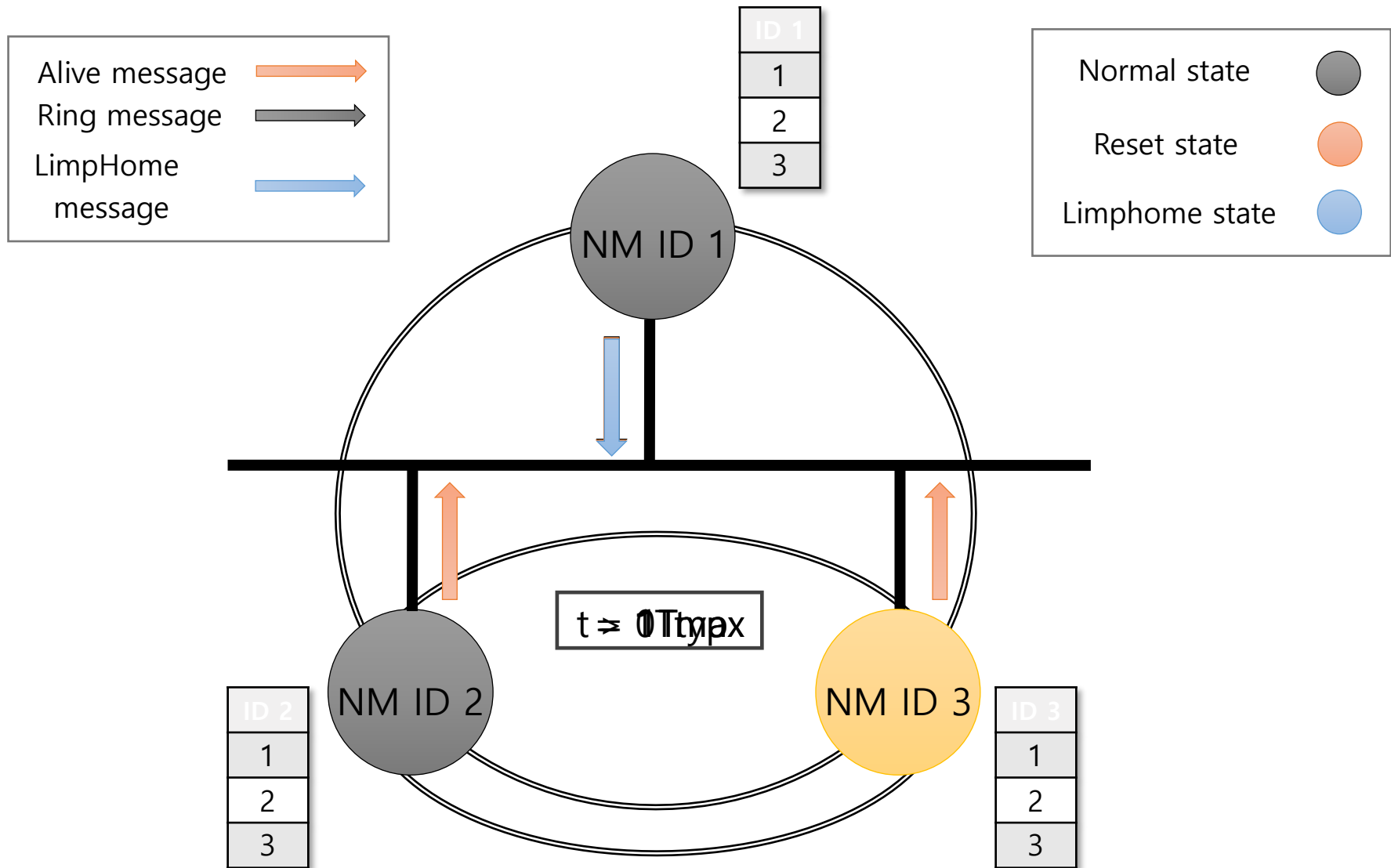
- 2 states of a monitored node
  - ✓ Node present
    - Specific NM message received (alive or ring)
  - ✓ Node absent
    - Specific NM message not transmitted during time-out
- 2 states of node itself
  - ✓ Present or not mute
    - Specific NM message received (alive or ring)
  - ✓ Absent or mute
    - Specific NM message not transmitted during time-out



# Logic ring

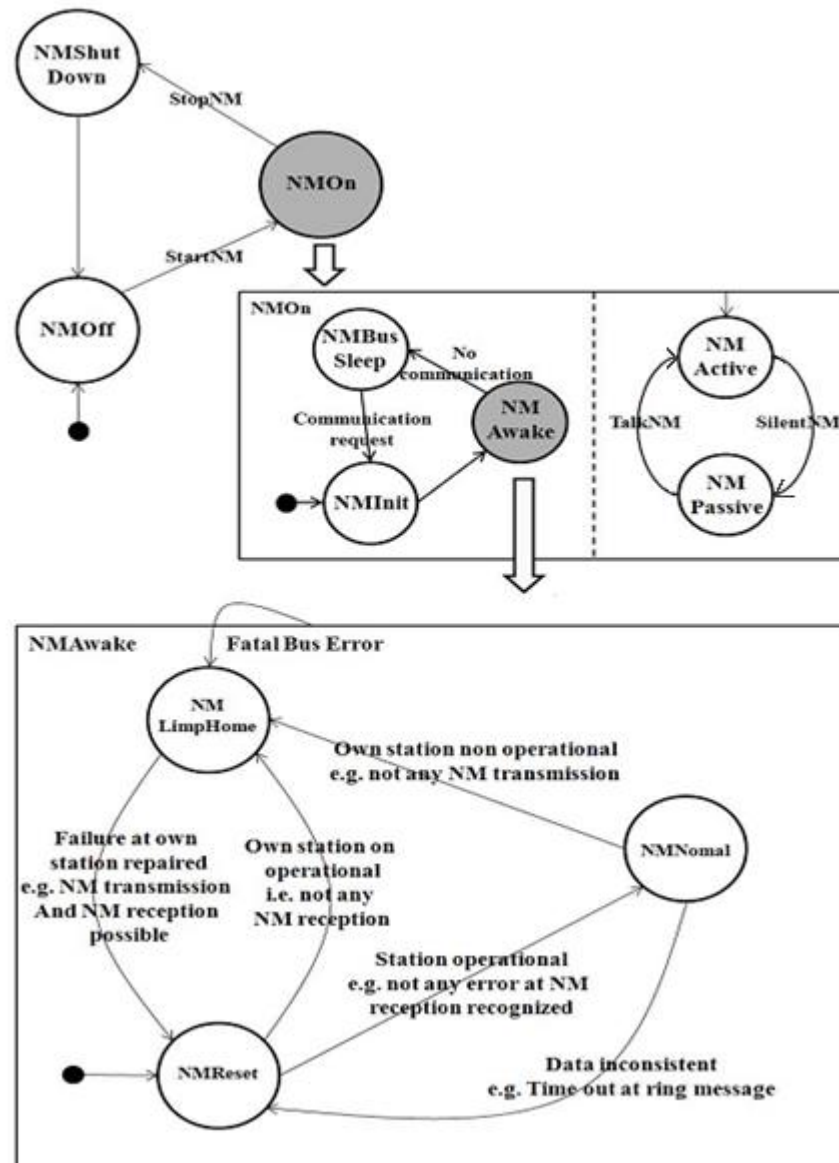


# Node Fault(limphone)



- NMON (NM is switched on)
  - ✓ NMInit: NM initialization
  - ✓ NMBusSleep : NM is in sleep mode
  - ✓ NMActive : NM communication enabled
  - ✓ NMPassive : NM communication disabled
  - ✓ NMAwake : Active state of the NM
    - NMReset : The operability of the own node is determined
    - NMNormal : Processing of direct node monitoring
    - NMLimpHome : Handling of failure in own node
- NMOff (NM is shut off)
- NMShutDown (Selective shut off of NM entity)

# Internal NM state

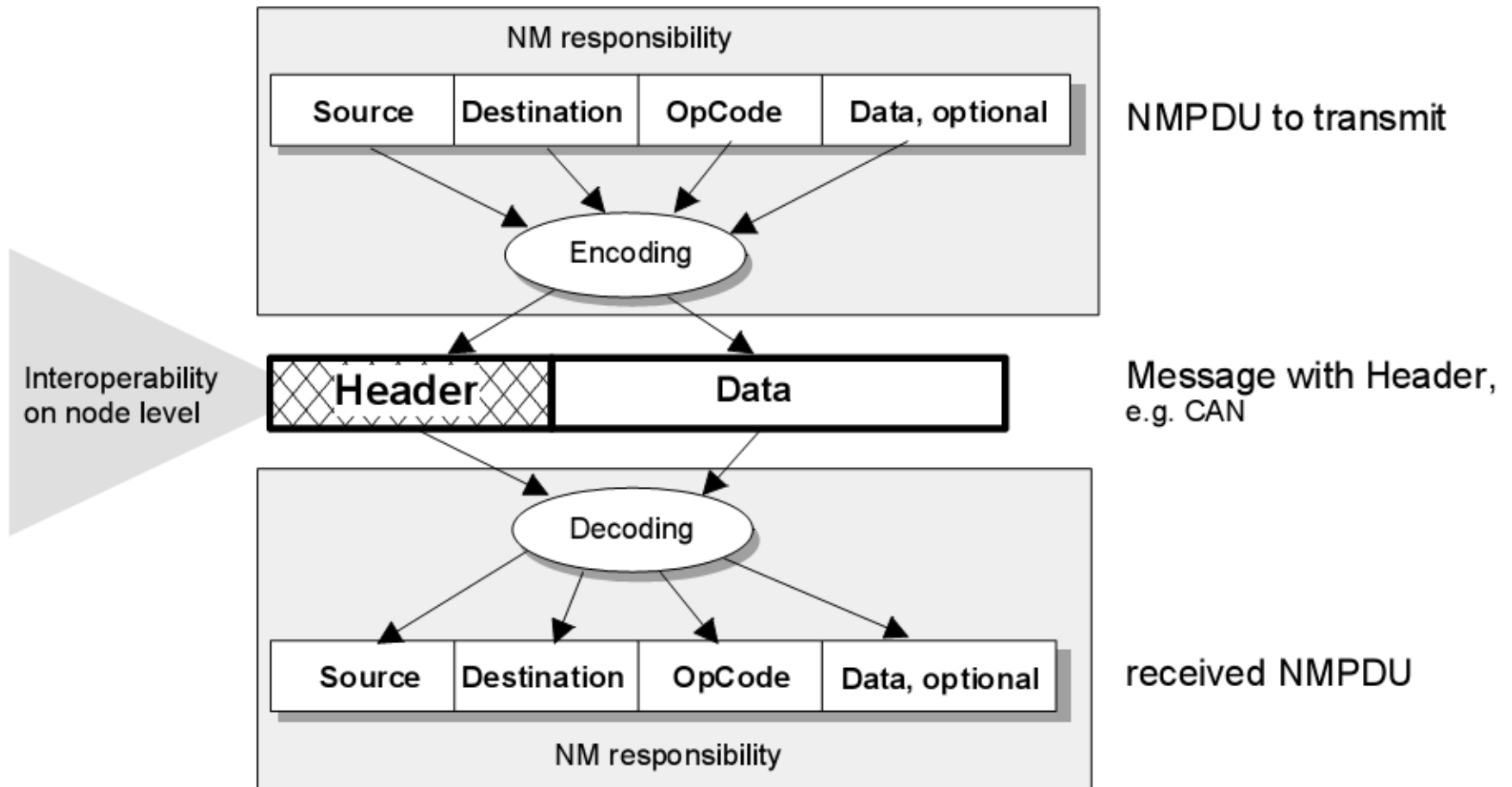


- NMPDU(Network Management Protocol Data Unit)
  - ✓ The NMPDU defined represents the OSEK NM data to be communicated in order to control NM
  - ✓ OSEK NM does not define network addresses
  - ✓ The parameter of NMPDU is dedicated to specific system design and therefore in the responsibility of the system developer
- Elements of NMPDU
  - ✓ NM address field : source ID, destination ID
  - ✓ NM control field : OpCode
  - ✓ NM data field(optional) : application specific data

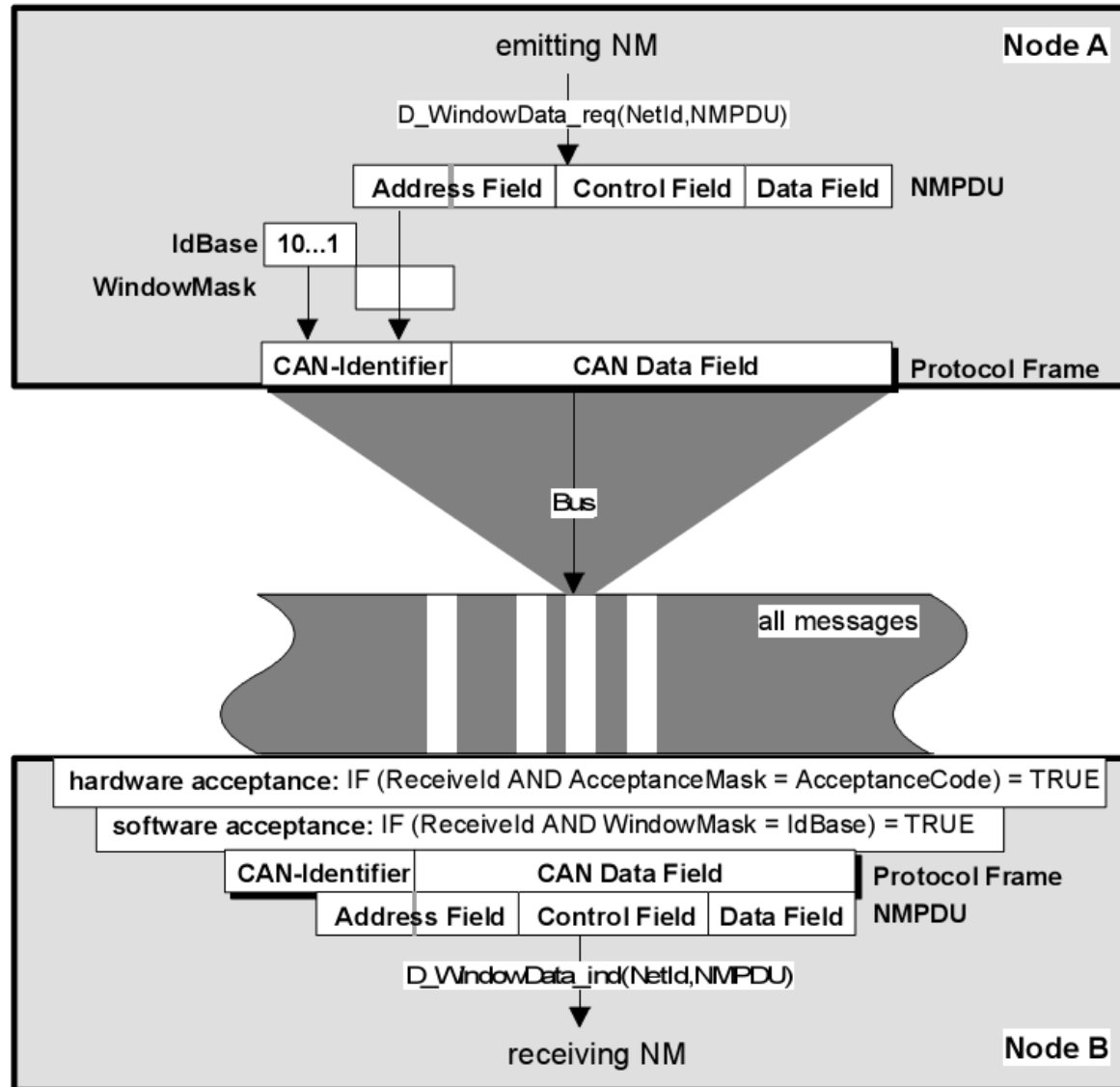
- Structure of NMPDU

Address Field		Control Field		Data Field
Source Id	Dest. Id	OpCode		Data
mandatory				optional
		res	Ring Message (4 types)	
			Alive Message (2 types)	
			Limp Home Message (2 types)	

- Encoding/decoding of the NMPDU



- CAN-example for mechanisms of a NMPDU

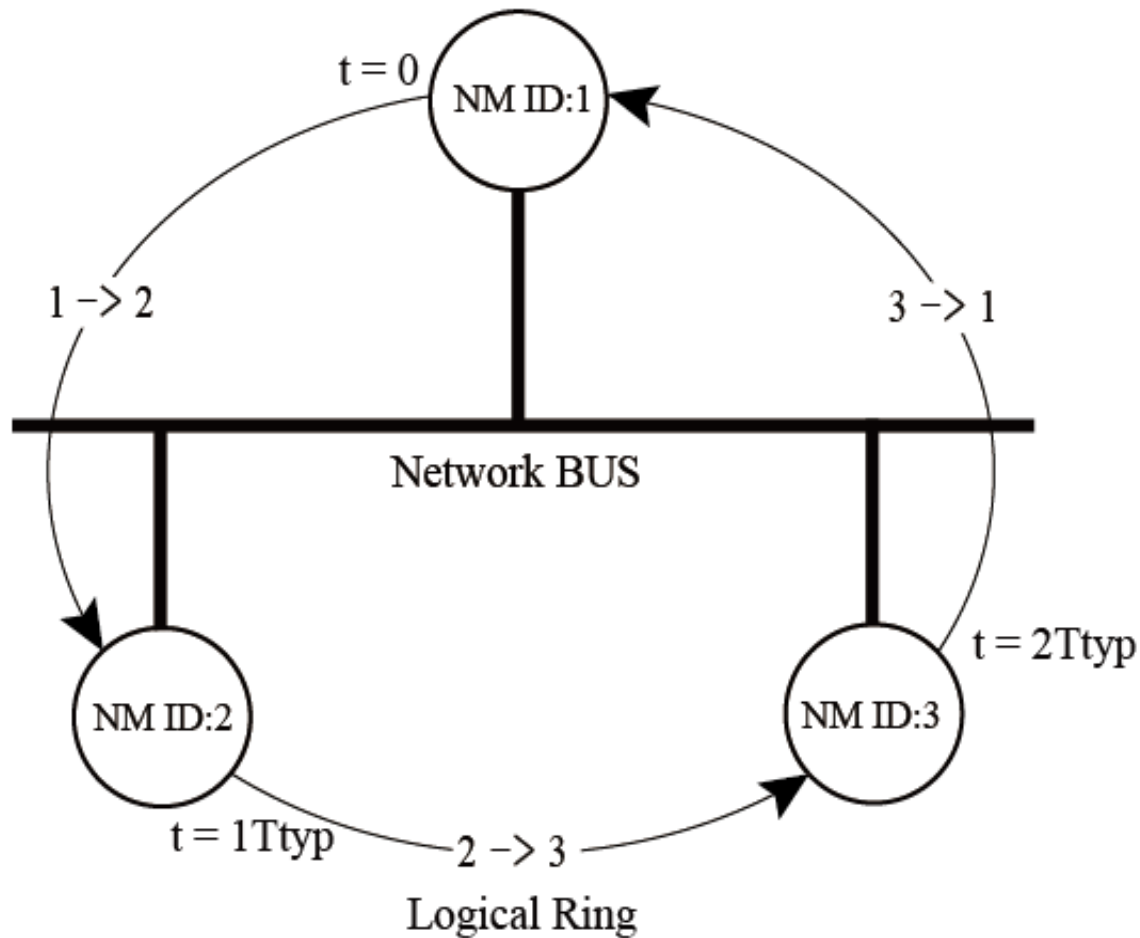




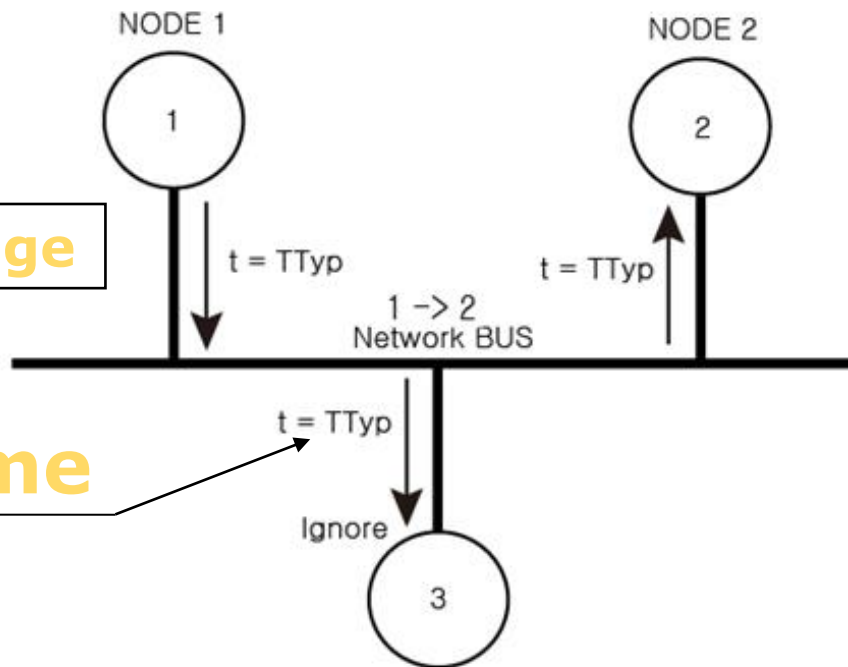
- NM parameters

NM Parameter	Definition	Valid Area
Nodeld	Relative identification of the node-specific NM messages	local for each node specific
TTyp	Typical time interval between two ring messages	global for all nodes
TMax	Maximum time interval between two ring messages	global for all nodes
TError	Time interval between two ring message with NMLimpHome identification	global all nodes
TWaitBusSleep	Time the NM waits before transmission in NMBusSleep	global all nodes
TTx	Delay to repeat the transmission request of a NM message if the request was rejected by the DLL(Dynamic Link Library)	local for each node specific

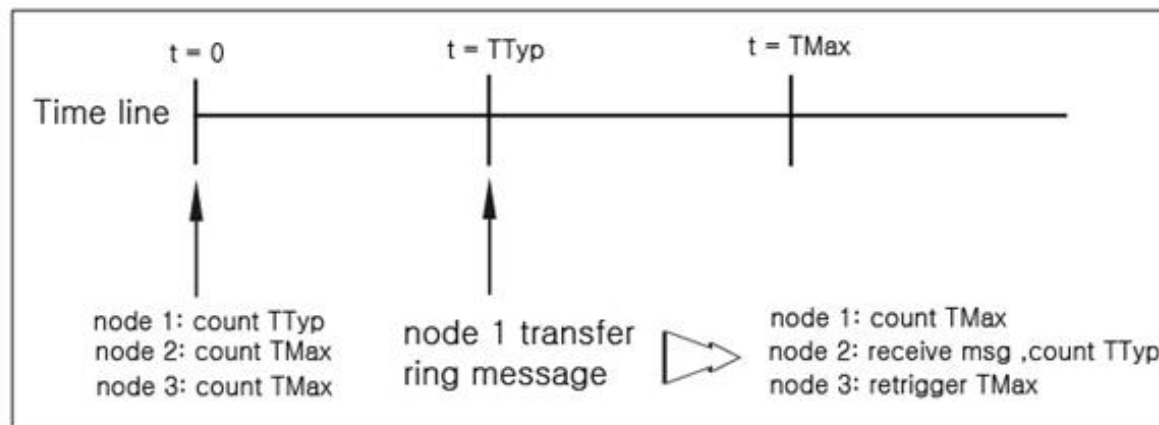
- Logic ring with ring message timing

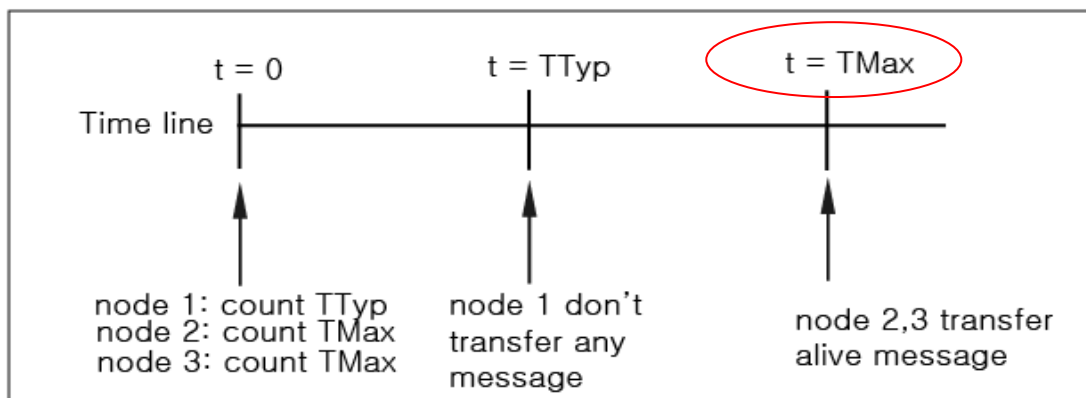
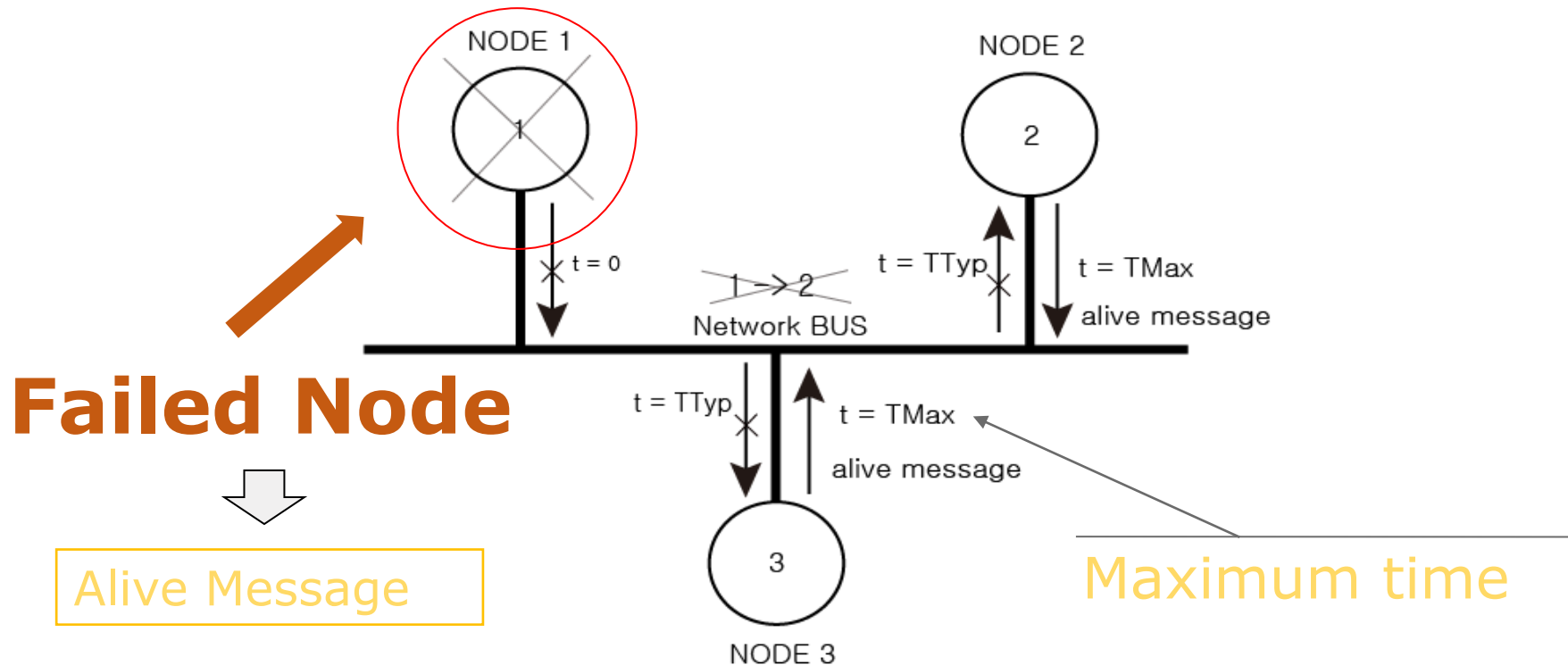


Ring message

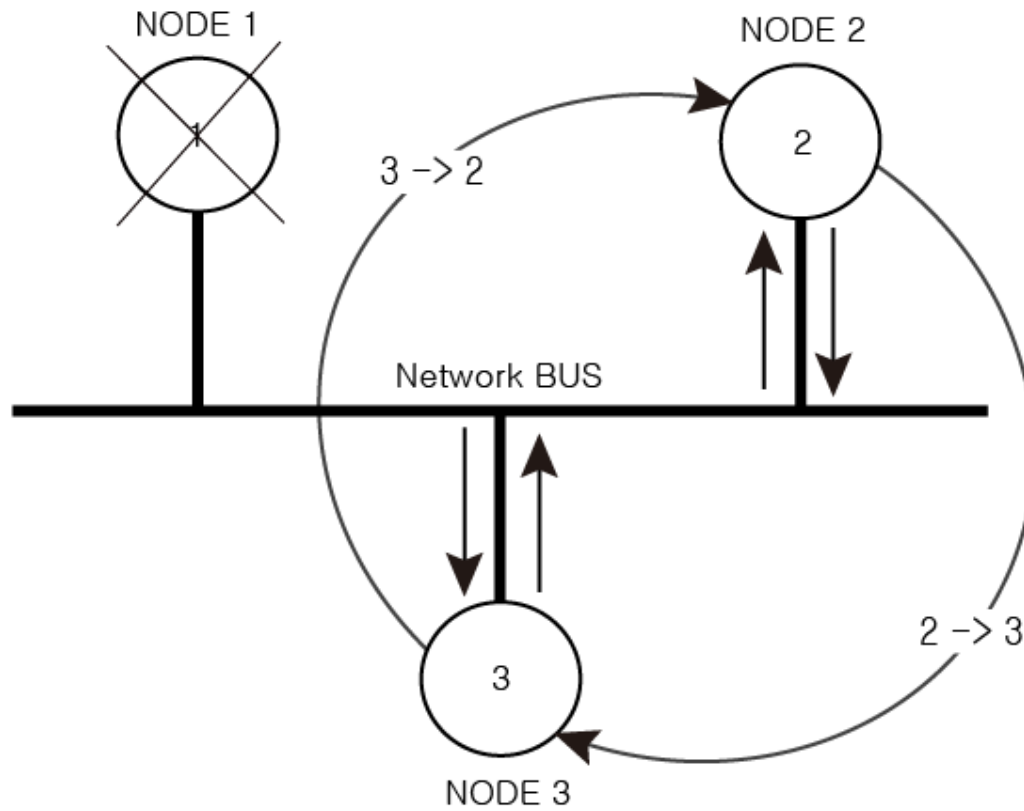


Typical Time

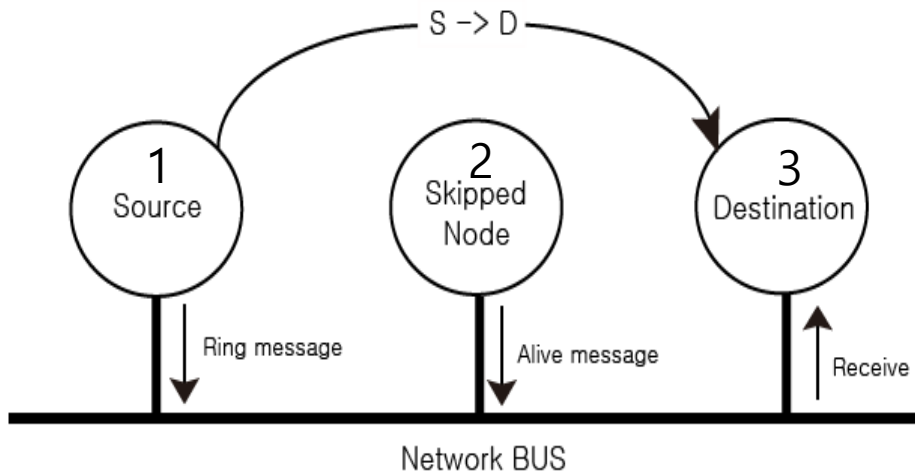




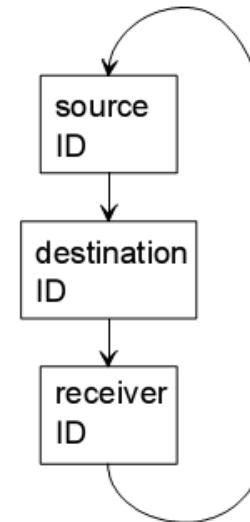
- New logic ring



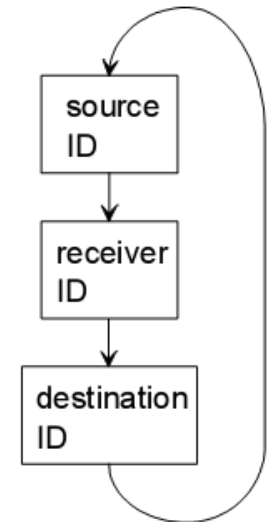
- Skipped node



receiver node  
was not skipped



receiver node  
was skipped



- Skipped node table

Numbering Order			Skipped or Not
Smallest num	Median num	Bigger num	
Source	Destination	Receiver	Not Skipped
Receiver	Source	Destination	Not Skipped
Destination	Receiver	Source	Not Skipped
Destination	Source	Receiver	Skipped
Receiver	Destination	Source	Skipped
Source	Receiver	Destination	Skipped

1. OSEK/VDX COM
  - 1.1. 개요
  - 1.2. 계층 구조
  - 1.3. Interaction Layer
  - 1.4. Message Reception
  - 1.5. Message Transmission
  - 1.6. Communication System Management
  - 1.7 Deadline Monitoring
2. OSEK/VDX NM
  - 2.1. 개요
  - 2.2. Direct OSEK NM
  - 2.3. Indirect OSEK NM



- Emitter states

- ✓ Node is not mute

- Specific application message transmitted

- ✓ Node is mute

- Specific application message not transmitted during a time-out

- Receiver states

- ✓ Node is present

- Specific application message received

- ✓ Node is absent

- Specific application message not received during a time-out

- Extended emitter states

- ✓ Node is not mute statically

- Specific application messages are transmitted

- ✓ Node is mute statically

- Specific application messages are not transmitted during a "long" time (several time-outs)

- Extended Receiver states

- ✓ Node is present statically

- Specific application messages are received

- ✓ Node is absent statically

- Specific application messages are not received during a "long" time (several time-outs)

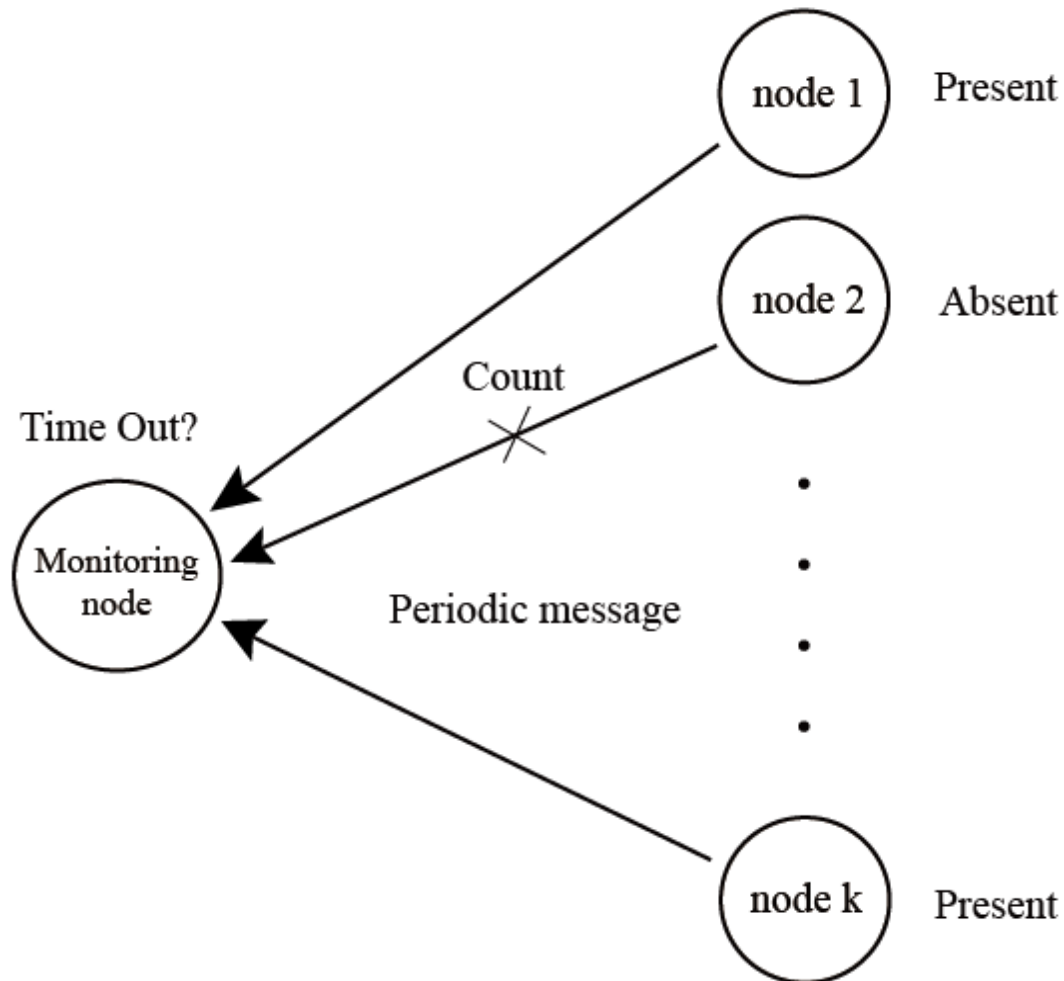
- **Transmission**

- ✓ Determination of the emitter states by using transmission monitoring scheme
- ✓ Transmission problems are detected by checking local confirmations related to transmissions of a periodic application frame chosen among those to be sent

- **Reception**

- ✓ Determination of the set of receiver states by using reception monitoring scheme
- ✓ Monitoring node checks the presence of all its source nodes by monitoring the reception of a chosen cyclic frame

- Monitoring mechanisms of Indirect NM



- Two time-out mechanisms

- ✓ All messages are monitored by one global time-out

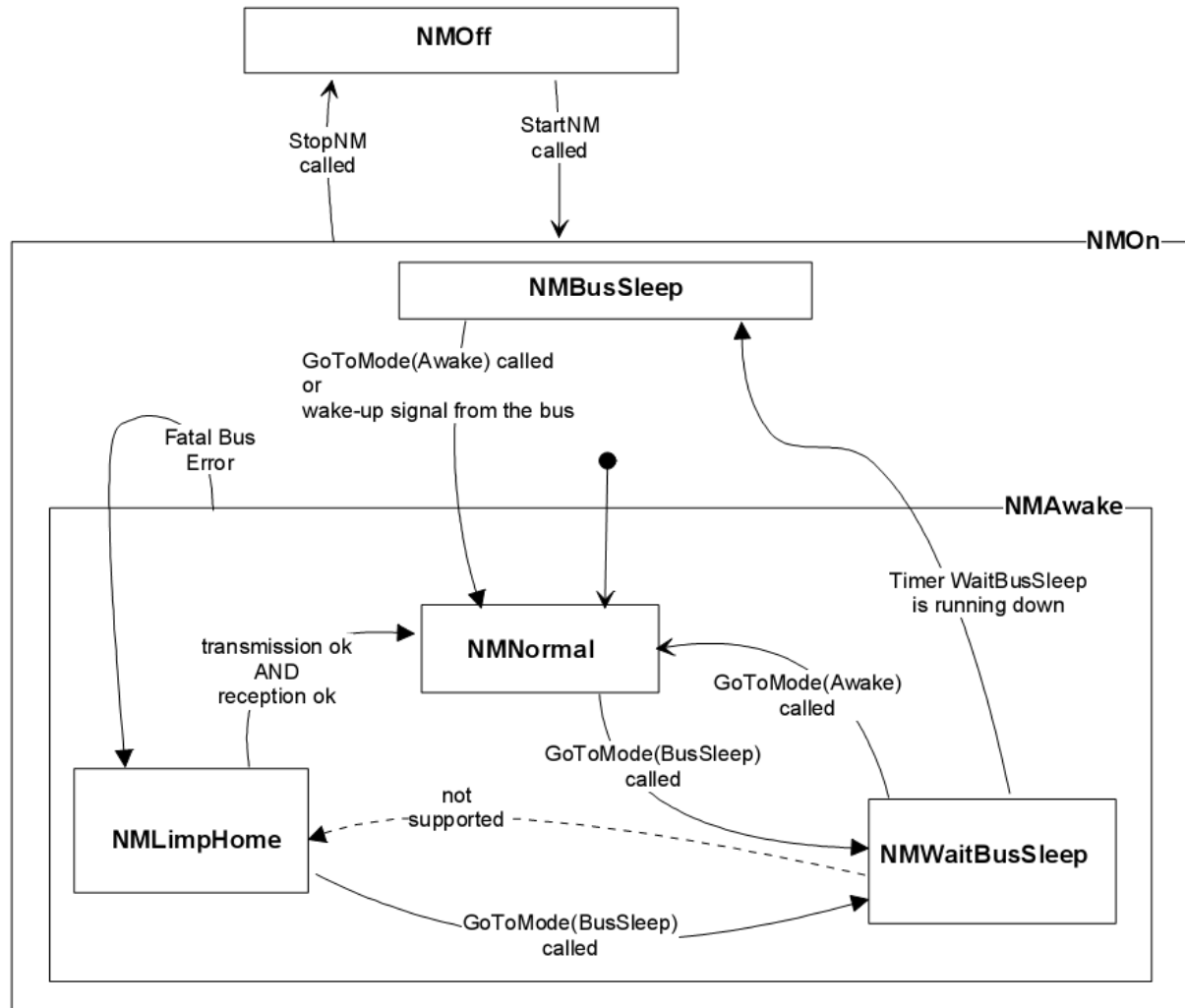
- The global monitoring time-out is located inside NM and is used as a observation
- The time-out require longest time among all the monitored application messages

- ✓ Each message is monitored by its own dedicated time-out

- Indirect NM uses "COM Deadline Monitoring" mechanisms to monitor dedicated application messages
- Time-outs are located at Interaction Layer level
- NM is informed dynamically by COM a message states which are correctly transmitted or received or time-out.

- NMON (NM is switched on)
  - ✓ NMBusSleep : NM is in sleep mode
  - ✓ NMAwake : Active state of the NM
    - NMNormal : Processing of direct node monitoring
    - NMLimpHome : Handling of failure in own node
    - NMWaitBusSleep : Synchronizing the network wide jump to the state BusSleep
- NMOff (NM is shut off)

- NM state of Indirect NM

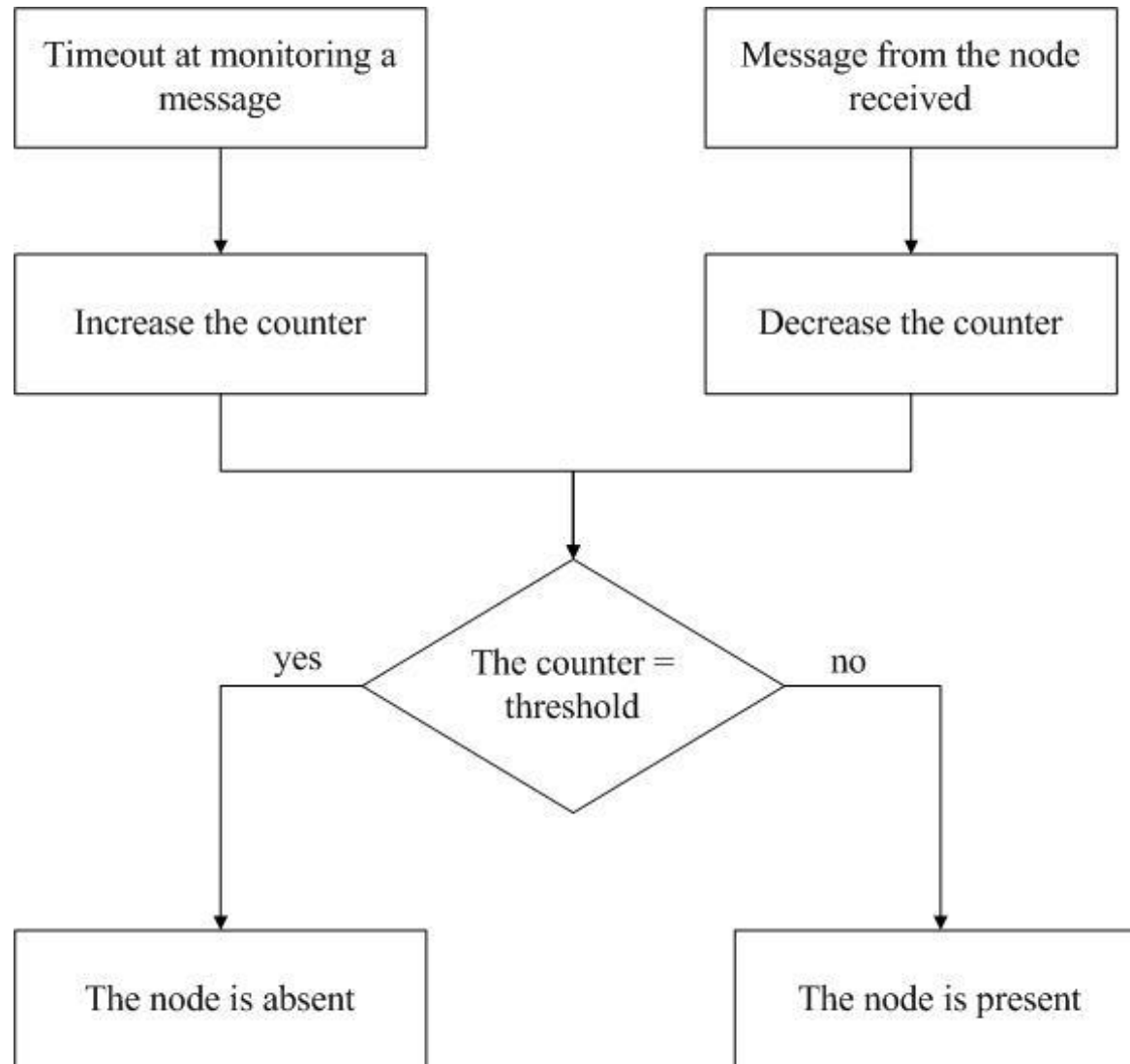


- Counter management

- ✓ When receiver receive the message, receiver decrease the counter
- ✓ When receiver does not receive the message and time-out expired, receiver increase the counter
- ✓ Receiver compare the counter with a threshold value then determine the emitter state
- ✓ OSEK NM sets the threshold to a constant value



- Mechanism of Indirect NM



- Time-out of Indirect NM

