

EtherNet/IP 기본 원리

(주)임베디드시스템코리아

EtherNet/IP 기본 원리

CIP 입문

CIP란 무슨 뜻입니까?

CIP 는 ***Common Industrial Protocol*** 을 말합니다

- DeviceNet (ODVA)
- ControlNet (CI)
- EtherNet/IP (ODVA/CI)

이러한 모든 3가지 기술들은 *공통적인* 다음을 공유합니다

- 객체(Object) 모델
- 통신 모델
- 7
- Device Profiles

CIP란 무슨 뜻입니까? (보충설명)

DeviceNet 은 가장 오래되고 널리 알려진 프로토콜입니다. 이것은 CAN (Controller Area Network)에 바탕을 두고 있습니다.

DeviceNet 은 1994년에 소개되었습니다. DeviceNet을 지원하는 모든 공급업체들을 위한 조직이 ODVA (Open DeviceNet Vendor Organisation)입니다.

ControlNet 은 CIP-패밀리의 두 번째 멤버이며 독점적인 물리 계층과 데이터 연결 계층에 기반을 두고 1996년에 소개되었습니다. 업체들은 CI (ControlNet International)로 조직되었습니다.

EtherNet/IP 는 CIP-패밀리의 가장 나중 멤버이며 2000년에 소개되었습니다. EtherNet/IP 는 산업용 이더넷에 기반을 두고 있으며 ODVA 와 CI의 협력 결과입니다.

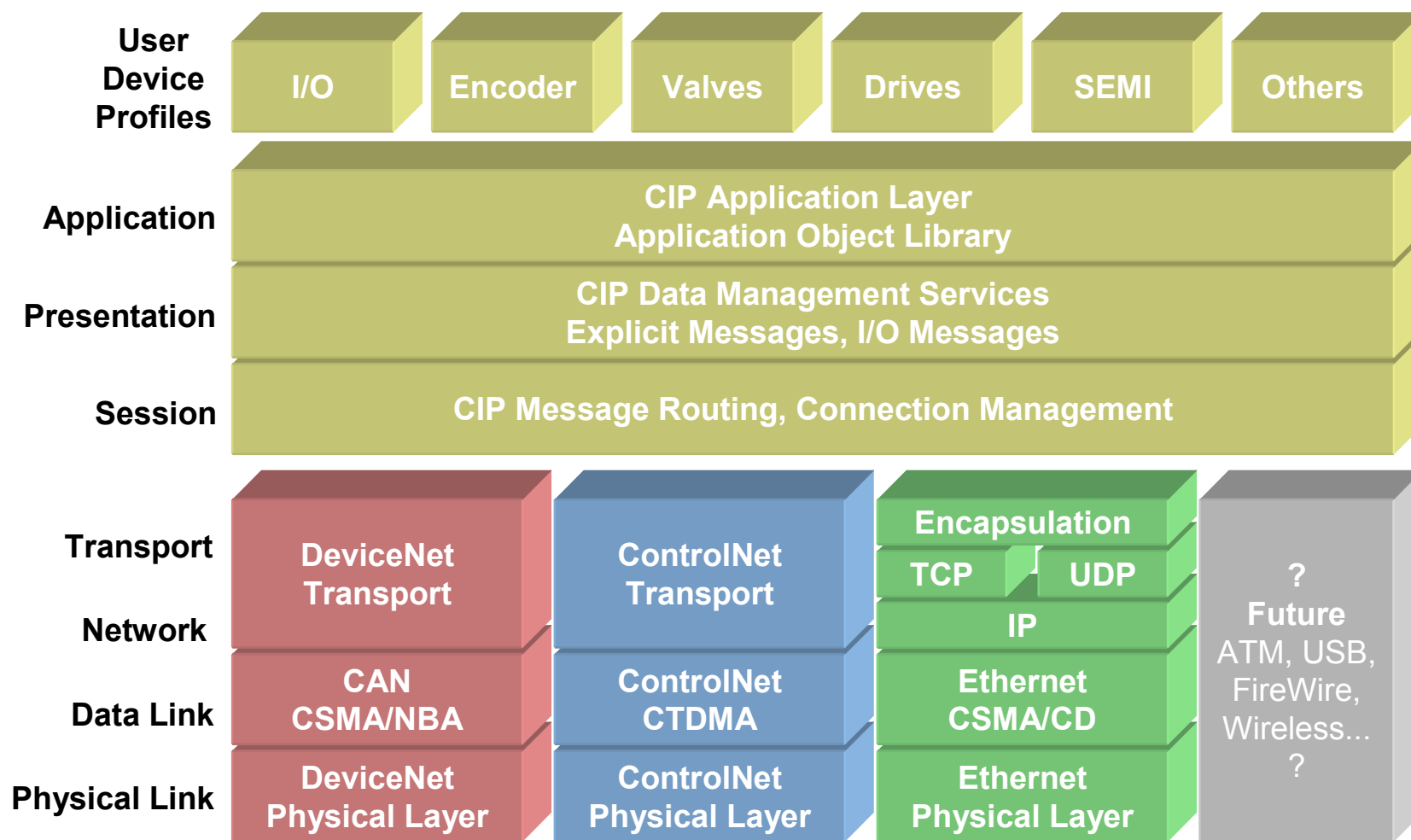
이 세 가지 기술들에 대해 과거에는 세 개의 별도 규격들이 있었습니다:

- DeviceNet, Volumes I und II
ControlNet, Volumes I und II
EtherNet/IP, Volumes I und II

ODVA 는 현재 규격의 새로운 구조를 공개했습니다. DeviceNet, ControlNet 과 EtherNet/IP의 예전 규격의 공통적인 - CIP - 파트가 수정되었으며 이를 기초로 새로운 규격이 만들어졌습니다. CIP 기술의 각각에 대해 별도의 규격이 생성되었습니다. 현재 4개의 규격들을 이용할 수 있습니다.

- Volume 1: The Common Industrial Protocol (CIP)
Volume 2: EtherNet/IP Adaptation of CIP
Volume 3: DeviceNet Adaptation of CIP
Volume 4: ControlNet Adaptation of CIP

CIP 계층 모델



CIP 계층 모델 (보충설명)

공통 산업용 프로토콜(Common Industrial Protocol)은 ISO/OSI-계층 모델에 따라 다음 계층들을 사용합니다:

Application (Layer 7)

- *Application Layer* 은 디바이스의 기능을 만드는데 사용되는 Object 라이브러리를 정의합니다.

Presentation (Layer 6)

- *Presentation Layer* 는 CIP에 사용되는 메시지와 데이터 관리를 정의합니다.

Session (Layer 5)

- CIP 는 연결-기반 프로토콜입니다. message routing 과 연결 관리는 *Session Layer* 에서 실현됩니다.

Device Profiles 은 공장 자동화에서의 사용을 위한 다양한 제품들을 만들도록 명시됩니다..

다음은 위한 기본 계층들은

- Transport (Layer 4)
Network (Layer 3)
Data Link (Layer 2)
Physical Link (Layer 1)

CIP의 세 가지 구현들에 대해 명확히 한정됩니다:

- DeviceNet
ControlNet
EtherNet/IP

CIP 구현 – DeviceNet

- DeviceNet은 Data Link 계층과 물리 계층으로서 CAN을 사용합니다
- CAN을 이용할 수 있는 표준 구성요소
- 모든 데이터는 CAN 메시지의 8 바이트 payload data에 맞춰져야 합니다
 - 더 큰 데이터 패킷들의 경우 단편화(Fragmentation)
- 효율적인 메시지 필터링과 간단한 연결 설치를 위한 Predefined Messages
- 애플리케이션 영역은 공장 자동화입니다:
 - I/O-Modules, Encoders, Drivers, Valves, HMI, Machine Control (PLC), ...

CIP 구현 – DeviceNet (보충설명)

DeviceNet 은 Rockwell Automation 과 Open DeviceNet Vendor Association에 의해 개발되었습니다. 이것은 EN 50325에서 표준화되었으며 CAN-프로토콜 (Controller Area Network)을 바탕으로 합니다. 이것은 CAN에 최적화 되었으므로 CAN 과 동작하며 추가 노력 없이는 다른 데이터 연결 계층에 맞출 수 없습니다. CAN의 물리 계층은 네트워크에서 메시지의 오류 없는 전송에 매우 중요합니다. DeviceNet의 규격은 데이터 속도, 케이블 종류, 케이블 길이와 물리 계층에 대한 인터페이스를 정의합니다.

일치여부(compliance)는 인증 과정에서 Physical Layer Test로 검사됩니다. CAN의 표준 요소들이 DeviceNet 제품들에서 사용됩니다 (CAN-Controller, uC with integrated CAN, CAN-Transceiver).

순수 CAN 메시지의 payload 데이터는 8 바이트로 제한됩니다. DeviceNet의 전송 계층에서의 단편화 프로토콜을 이용하여,

네트워크에서 8 바이트 이상의 payload 데이터를 보내는 것이 가능합니다. CIP를 이용해서 데이터의 최대 길이는 64 kbyte입니다.

Common Industrial Protocol 은 CAN에서 사용된 것과 동일한 원리로, 네트워크에서 데이터를 교환하는데 Producer-Consumer-Model 을 사용합니다.

CAN 은 모든 하드웨어 핸드셰이크를 지원하지 않으므로, 네트워크에 있는 모든 노드들은 실시간에서 소비 메시지들을 여과시킬 수 있어야 합니다. 이것은 하드웨어-filtering (CAN 컨트롤러에서) 또는 소프트웨어-filtering (마이크로컨트롤러에서)을 통해 실행될 수 있습니다. 후자는 고성능의 마이크로컨트롤러를 필요로 합니다.

느린 8비트 마이크로컨트롤러 기능을 가진 소형 제품들을 위해, DeviceNet은 매우 효과적인 메시지 코딩을 명시해서 하드웨어-필터링이 단 한 개의 수신 버퍼와 Mask-And-Match 성능을 지원하는 CAN-컨트롤러들(예전 Philips 82C200 같은) 과 사용될 수 있습니다. 이러한 종류의 메시지는 DeviceNet에서 *Predefined Master/Slave Connection Set* 으로 알려져 있습니다. 이것들은 PLC (*master*) 가 네트워크에서 여러 개의 노드들 (*slaves*) 과 데이터를 교환하는 Master/Slave 시스템에서 사용됩니다.

CIP 구현 – DeviceNet

특징

- 네트워크에서의 노드 수
- 데이터 속도
- 네트워크 위상
- 케이블 종류
- 커넥터
- 케이블 길이
- 24 V 전원 장치

CIP 구현 – DeviceNet (보충설명)

DeviceNet 네트워크는 최대 64개의 노드를 지원합니다. 이 제약은 CAN 식별자에서 선택된 어드레싱 코드의 결과입니다 (6비트가 MAC-ID에 사용됩니다).

3 Data Rate 가 정의됩니다: 125 kbps, 250 bps 와 500 kbps.

네트워크 위상은 trunk line, tees, taps 과 drop 라인이 포함된 *line* 구조를 따릅니다. Star 와 tree 같은 다른 위상들도 가능하지만 제약이 있습니다.

케이블은 CAN 신호와 전원 공급을 위한 별도 쌍이 있는 차폐 연선 (shielded twisted pair)입니다. 다양한 케이블 종류들이 정의됩니다. 예. 두꺼운 케이블, 가는 케이블, 편평한 케이블. 커넥터들은 다양한 애플리케이션들을 지원하며 Open Style, Micro Style oder Mini Style로 이용할 수 있습니다. 또한 클램프를 이용한 매체 부착도 가능합니다.

네트워크에서 사용되는 케이블 길이는 데이터 속도와 케이블 종류에 따라 달라집니다. 두꺼운 케이블의 경우 다음 값들이 지정됩니다: 125 kbps 에서 500m, 250 kbps 에서 250m, 500 kbps에서 100m. 가는 케이블을 사용할 때는, 최대 케이블 길이가 데이터 속도와 상관없이 항상 100m 입니다. 드롭 라인의 최대 길이 또한 지정됩니다.

DeviceNet 노드는 네트워크를 거쳐 24볼트가 공급될 수 있습니다. 케이블 종류와 케이블 길이에 따라 최대 8A가 공급될 수 있습니다.

구성과 진단이 DeviceNet 네트워크를 거쳐 실행되며, 추가 인터페이스는 필요하지 않습니다.

CIP 구현 – ControlNet

- ControlNet은 독자적인 Data Link 계층과 물리 계층을 사용합니다
- 특수 하드웨어 필요
- I/O 데이터 크기 최대 510 byte
- 높은 결정력(deterministic)과 반복 가능
- 애플리케이션 영역은 빠른 실시간 제어입니다
 - Drives, Welding Control, Motion Control, complex I/O-Systems, ...

CIP 구현 – ControlNet (보충설명)

ControlNet은 Rockwell Automation und ControlNet International 에 의해 개발되었습니다. 이것은 EN 50170에서 표준화됩니다.

ControlNet은 자체의 특정한 Data Link 계층과 물리 계층을 사용합니다. 네트워크 접속은 Concurrent Time Domain Multiple Access (CTDMA)로 불리는 시간-분할 알고리즘에 의해 제어되는데, 각 네트워크 간격에서 전송하는 노드의 기회를 규제합니다. 구성 가능한 가장 빠른 NUT 는 2 ms 입니다. 정보가 Time-critical 인 것은 NUT의 예정된 부분 동안에 전송됩니다. 시간 제약없이 전달될 수 있는 정보 (configuration data 같은) 는 NUT의 예정되지 않은 부분 동안에 전송됩니다.

I/O 데이터의 최대 510 바이트가 ControlNet과 교환될 수 있습니다.

ControlNet 은 높은 결정력과 반복이 가능하므로, 선호되는 애플리케이션 영역은 빠른 실시간 제어입니다.

CIP 구현 – ControlNet

특징

- 네트워크에서의 노드 수
- 데이터 속도
- 네트워크 위상
- 중복(Redundancy)
- 케이블 종류
- 커넥터
- 케이블 길이
- 구성과 진단 인터페이스

CIP 구현 – ControlNet (보충설명)

ControlNet 네트워크는 최대 99개의 노드를 지원합니다.

데이터 속도는 5 Mbps 입니다.

네트워크 위상은 trunk 라인, tees, taps 과 drop 라인들의 라인 구조를 따릅니다. Star 와 tree 같은 다른 위상들은 repeater를 사용하여 쉽게 실현될 수 있으므로 ControlNet 네트워크는 최대 25 km 까지의 큰 범위에 도달할 수 있습니다.

ControlNet 은 여분의 케이블 설치를 지원합니다.

케이블로는 쿼드 차폐 RG-6 동축 케이블 (quad shielded RG-6 coaxial cables) 들이 일반 BNC 커넥터와 사용됩니다. 광섬유 전송 또한 제공됩니다.

ControlNet 네트워크에서의 케이블 길이는 연결된 노드 수에 따라 달라집니다. 48 개의 노드들이 있을 때는 250m 로 제한되며 2개의 노드에서는 최대 1km 까지 증가될 수 있습니다. 네트워크를 끊지 않고도 추가 인터페이스 **ControlNet NAP** (Network Access Port) 를 통해 구성과 진단이 실행될 수 있습니다.

NAP은 이를 위해 RS422 interface (RJ45 jack)을 제공합니다.

CIP 구현 – EtherNet/IP

- EtherNet/IP는 Ethernet (IEEE 802.3)에 기초를 두고 있습니다
 - 판매용 기성품 (COTS)
 - 산업용 이더넷
- 전송 계층으로 TCP/IP 와 UDP/IP 를 사용합니다.
- I/O 데이터 크기 최대 64 kbyte
- HTTP, FTP, SMTP같은 다른 프로토콜들은, 한 네트워크에서 동시에 사용될 수 있습니다
- 애플리케이션 영역은 공장 작업장에서부터 기업 네트워크에 이르는 자동화 네트워킹입니다

CIP 구현 – EtherNet/IP (보충설명)

EtherNet/IP 는 ODVA 와 CI 에 의해 공동으로 개발되었습니다.
이것은 IEEE 802.3 에 따라 Ethernet에 기반을 두고 있습니다.

이것은 상업용 오프 더 쉘프와 산업용 이더넷 요소를
사용합니다.

EtherNet/IP 는 CIP 메시지를 기본 프로토콜 TCP/IP 와 UDP/IP
에 대응시키는 캡슐화 (encapsulation) 프로토콜을 제공합니다.

HTTP (Hypertext Transfer Protocol), FTP (File Transfer
Protocol) 또는 SMTP (Simple Mail Transfer Protocol) 같은 다른
잘 알려진 인터넷-프로토콜들은 하나의 네트워크에서
EtherNet/IP 와 병렬로 사용될 수 있습니다.

CIP 구현 – EtherNet/IP

특징

- 네트워크에서의 노드 수
- 데이터 속도
- 네트워크 위상
- 중복(Redundancy)
- 케이블 종류
- 커넥터
- 케이블 길이

CIP 구현 – EtherNet/IP (보충설명)

EtherNet/IP 네트워크에 연결된 노드들의 수는 단지 내재하는 이더넷 프로토콜 (MAC-Address)과 인터넷 프로토콜 (IP-Address)의 주소 배정 공간에 의해 제한됩니다.

두 개의 데이터 속도들이 정의됩니다: 10 Mbps 와 100 Mbps.

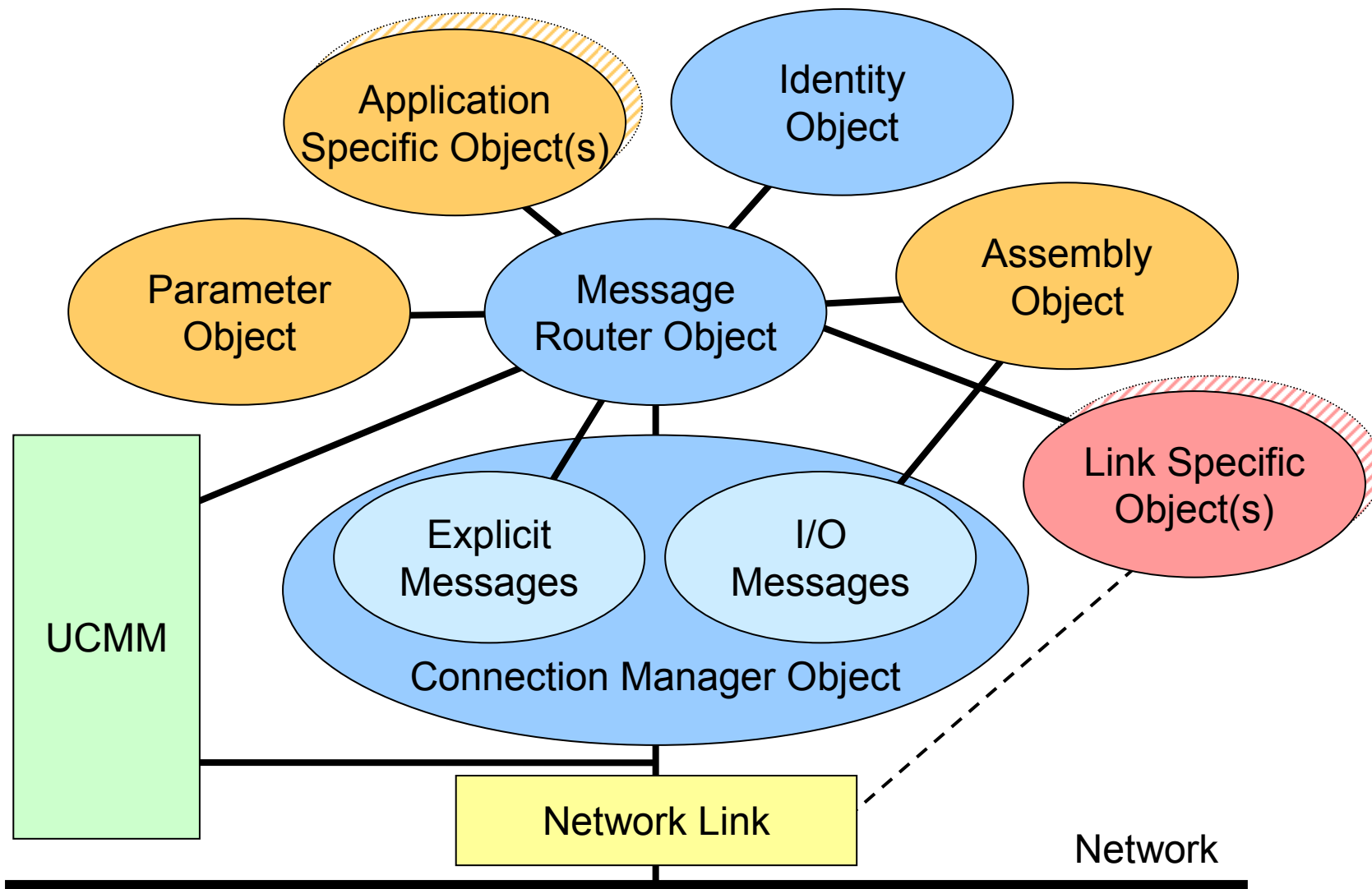
네트워크 위상은 허브, 스위치와 라우터를 사용하여 **star** 와 **tree** 위상을 따릅니다.

케이블 설치에 차폐된 꼬임 쌍 (shielded twisted pair) 케이블 또는 비차폐된 꼬임쌍 (unshielded twisted pair) 케이블들이 사용됩니다.

커넥터로는 RJ-45 또는 M12-4 Type „D“ 가 지정됩니다.

두 말단 사이의 케이블 길이는 내재된 이더넷 계층에 의해 100m 까지 제한됩니다. 이것은 매우 큰 네트워크를 계획할 때 고려되어야 합니다.

객체 모델



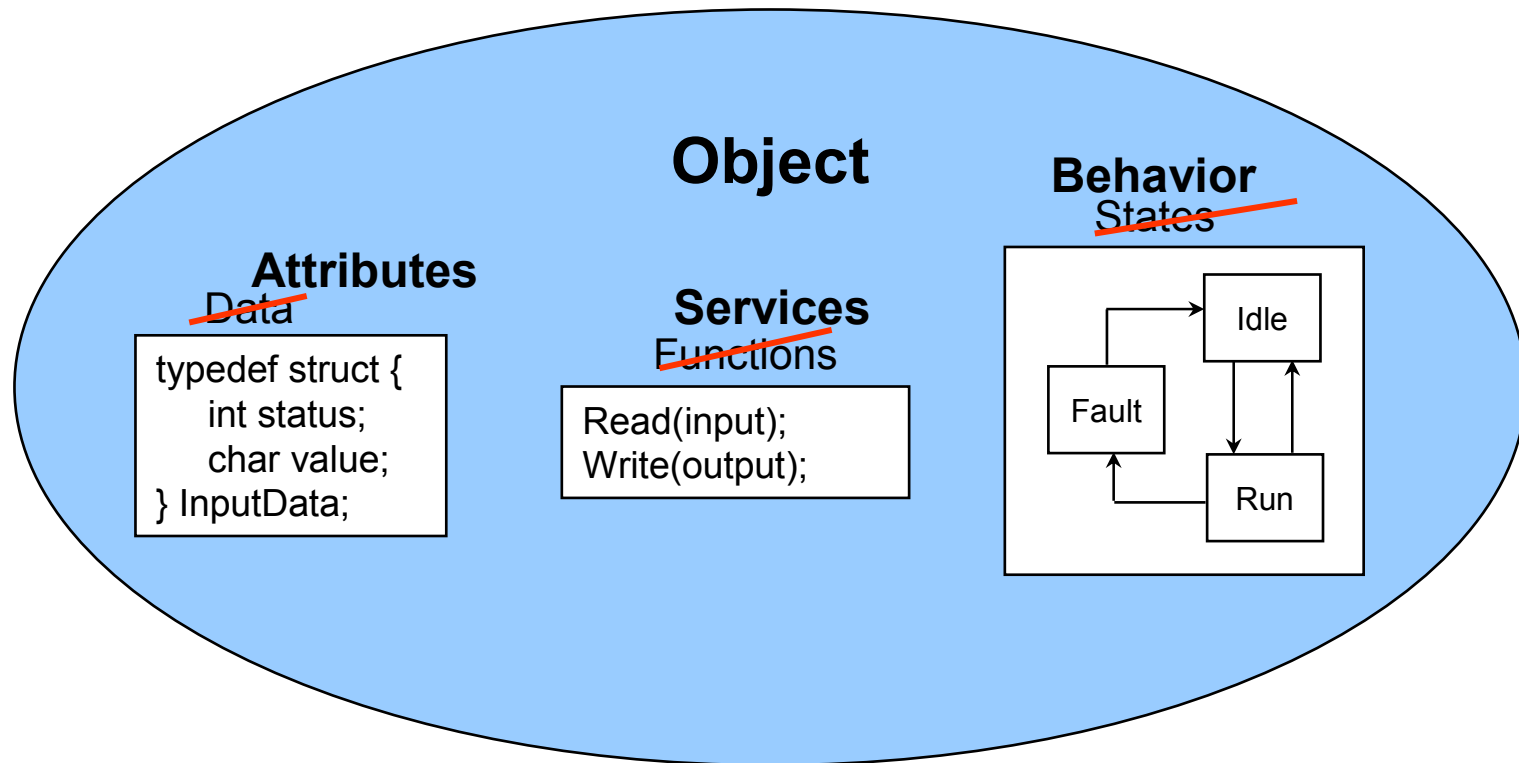
객체 모델 (보충설명)

CIP 는 디바이스의 통신과 애플리케이션을 서술하기 위해 추상적인 *Object Model* 을 사용합니다. 디바이스 내부 데이터, 기능, 상태들은 오브젝트에 의해 기술됩니다. 각 CIP 디바이스들은 오브젝트의 컴파일(compilation)을 제공하며, 선택은 Device profile 에 명기되어 있습니다.

공급업체들은 CIP common object를 사용할 수 있으며, 그들은 자신들의 필요에 맞게 CIP common object를 확장하거나 새로운 오브젝트를 생성할 수 있습니다.

오브젝트 모델링 목표는 정보처리상호운용성 (*Interoperability*) 과 상호교환성(*Interchangeability*)을 얻기 위함입니다.

객체 모델

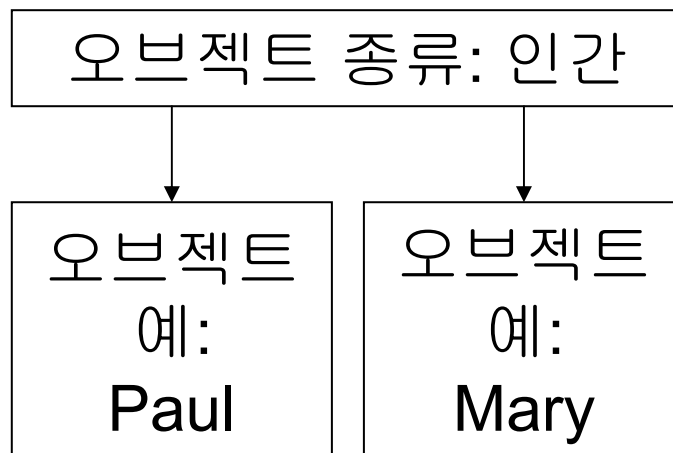


객체 모델 (보충설명)

전통적인 소프트웨어 설계는 데이터 구조(data structures), 기능(functions), 상태(states)를 사용합니다. CIP 오브젝트 모델 유닛들 이러한 것들은, 용어를 일치시켜서 이것을 *object* 라 부릅니다:

- data 와 structure 들은 *attributes* 로 명명됩니다
- function 들은 *services* 로 명명됩니다
- state 들은 *behaviour* 로 명명됩니다

객체 모델의 예



속성	
몸무게: 65 kg	몸무게: 55 kg
키: 1.8 m	키: 1.7 m
나이: 28 years	나이: 35 years

객체 모델의 예 (보충설명)

앞 페이지의 예는 두 사람에게서 오브젝트 모델의 구현을 보여줍니다. 여기서 우리는 오브젝트 클래스를 *human* 으로 정의합니다.

각 사람에 대해 오브젝트 클래스의 사본(copy)이 생성됩니다. 이 사본을 **object instance** 라 부릅니다. 각 **instance** 는 그 사람을 서술하는 속성들의 집합을 포함합니다. 여기 우리의 예에서는 세 가지 속성들이 구현되었습니다.

CIP 메시지 유형

CIP는 연결 기반 프로토콜입니다

CIP는 Messaging Connections을 바탕으로 합니다

- 명시적 Messaging Connections
 - 공통 데이터 교환
 - 지점 대 지점(Point-to-Point) 연결
 - 타겟 오브젝트에서 Service 수행
 - 클라이언트/서버 = 요청/응답
- I/O Messaging Connections
 - I/O 데이터 교환
 - Producer/Consumer Model

CIP 메시지 유형 (보충설명)

CIP 는 연결 기반 프로토콜이며, 네트워크에서 데이터를 교환하기 위한 두 가지 다른 방법들을 제공합니다:

명시적 메시징 연결 (Explicit Messaging Connections)

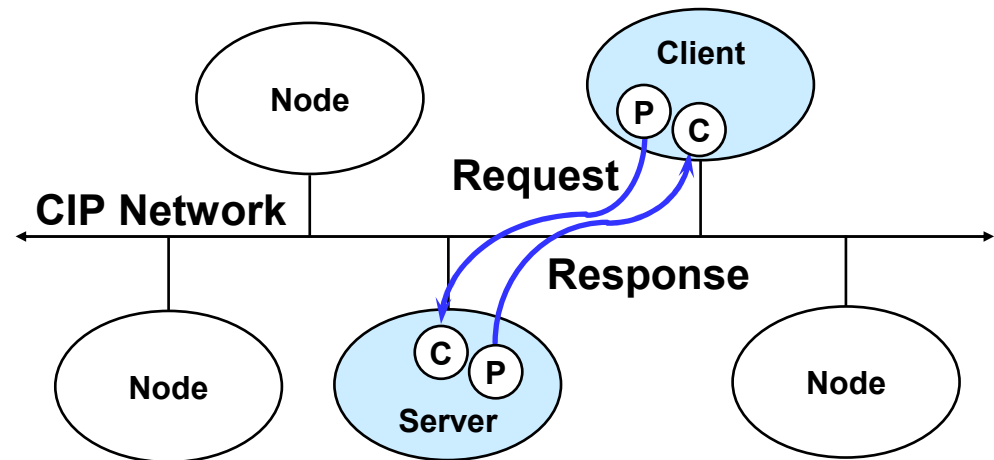
Explicit Messaging Connections은 네트워크에서 구성 데이터, 진단 데이터 또는 관리 데이터를 교환하기 위한 두 장치들 간의 공통적인 통신 경로로 사용됩니다. 이 데이터의 의미는 전송된 메시지를 해석하여 결정되어야 하며, 이것은 메시지에 *명시적으로* 포함됩니다. 명시적 메시징은 항상 지점 대 지점 (Point to Point) 연결을 사용하며 클라이언트가 서버에 있는 타겟 오브젝트에서 서비스를 요청합니다. 서버에 있는 오브젝트는 그 서비스를 실행하며 클라이언트로 응답을 되돌려 보냅니다.

I/O 메시징 연결

I/O 메시징 연결들은 Producer/Consumer 모델에 따라 한 개의 생산 오브젝트와 하나 또는 그 이상의 소비 오브젝트들 사이에서 I/O 데이터 또는 애플리케이션 데이터를 교환하는데 사용됩니다. 이 데이터의 의미는 통신 경로의 일부이며, 따라서 I/O 연결들은 암시적(*implicit*) 연결이라 명명됩니다. 데이터를 사용하는 모든 말단(endpoint)들은 이것이 메시지의 일부가 아니므로 데이터의 의미에 대한 지식을 갖고 있어야만 합니다.

명시적 메시지 (Explicit Messages)

- Producer/Consumer 모델
- 지점 대 지점
- 클라이언트/서버 관계
- 클라이언트는 서버에서 서비스를 요청
- 연결 또는 비연결



명시적 메시지 (보충설명)

명시적 메시지 (*Explicit messages*)들은 **Producer/Consumer Model**을 사용합니다. 이것들은 항상 두 말단 사이에서는 지점 대 지점 (*Point-to-Point*) 연결입니다. 한 말단은 클라이언트이며 다른 하나의 말단은 이 메시지를 위한 서버입니다. 클라이언트는 서버에서 서비스를 요청합니다.

Explicit Message Client 는 서버로 요청 메시지를 전송(생산)하고 그 응답 메시지를 수신(소비)합니다. *Explicit Message Server* 는 클라이언트로부터 요청 메시지를 수신(소비)하고 응답 메시지를 전송(생산)합니다.

생산과 소비 말단들은 오브젝트에 연결됩니다. 서버에서 소비 말단은 항상 메시지 라우터에 연결됩니다.

명시적 메시징을 이용해서 클라이언트는 서버에서 서비스를 실행합니다. 예를 들면, 데이터 읽기 (*service Get_Attribute_Single*). 서비스의 의미와 서버에서 그 서비스를 실행하는 오브젝트의 주소는 메시지의 일부입니다.

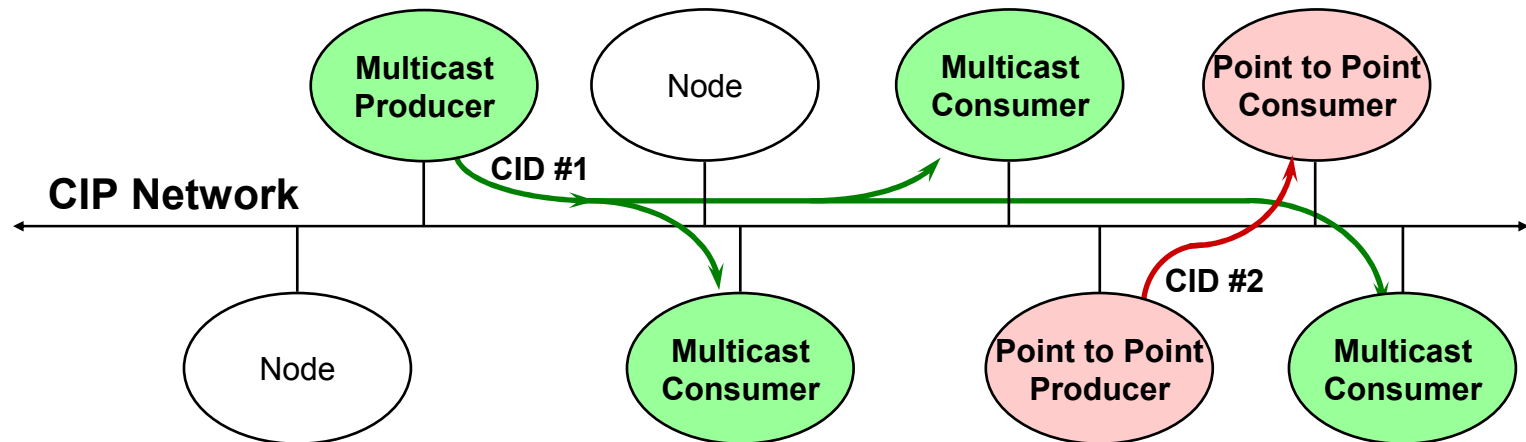
CIP 디바이스는 명시적 메시지 클라이언트와 명시적 메시지 서버가 동시에 될 수 있습니다. 또한 하나의 디바이스에서 다중의 명시적 메시징 연결들도 가능합니다.

명시적 메시지들은 연결을 통해 (*connected*) 그리고 무-연결 (*unconnected*)로도 전송될 수 있습니다.

메시지를 전송하기 위한 내재된 계층들에 대한 **CIP** 명시적 메시지의 적용은, 예를 들어 **CAN** 또는 **TCP/IP**를 사용하여, **CIP** 구현의 일부이며 **DeviceNet**, **ControlNet** 과 **EtherNet/IP**에 따라 달라집니다.

I/O 메시지

- Producer/Consumer 모델
- 연결이 필요합니다
- Connection ID (CID)
- 연결은 애플리케이션 데이터의 교환을 제어합니다 (Cyclic, 상태 변경 등.)



I/O 메시지 (보충설명)

I/O messages 역시 **Producer/Consumer Model** 을 사용합니다. *I/O* 메시지들은 항상 **connected** 입니다.

I/O Client 는 *I/O Server*에 대한 *I/O* 연결을 개시합니다. 특별한 식별자 (**Connection ID**)가 *I/O* 연결에 할당됩니다. 네트워크에 있는 소비 노드들은 그들에게 관련된 메시지들을 여과시키는데 이 식별자를 사용합니다. *I/O* 연결에 있는 말단은 생산자 또는 소비자가 될 수 있습니다.

client 와 *server* 용어는 네트워크에서의 두 개 말단 사이의 관계를 설명하는데 사용됩니다. *producer* 와 *consumer* 용어는 그것들과 상관이 없습니다. **CIP**를 이용하여 네트워크에서 클라이언트는 소비자로서 서버는 데이터의 생산자로 *I/O* 연결을 설정할 수 있습니다.

생산과 소비 말단들은 오브젝트에 연결됩니다. 이러한 오브젝트들에 대한 경로를 **Connection Path** 라고 부릅니다.

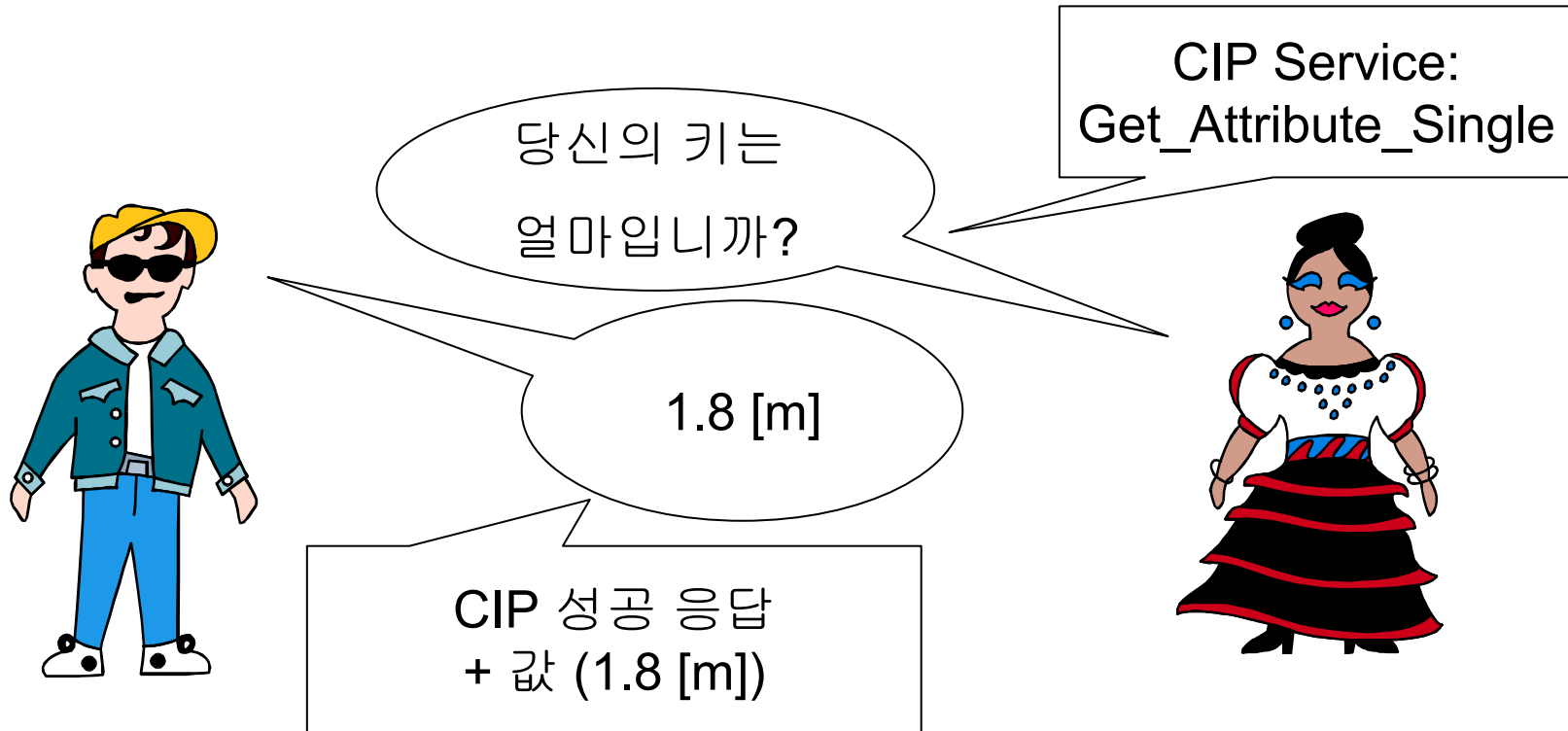
이 연결들은 그들이 묶여있는 오브젝트들과는 상관없이 네트워크에서 데이터를 생산하는 것을 책임집니다. **CIP** 로 애플리케이션과 통신 간의 명확한 분리가 실현됩니다.

I/O 연결들은 어떠한 프로토콜 오버헤드 없이 데이터를 전송합니다. 연결의 말단은 데이터의 의미에 대한 지식을 갖고 있어야 합니다. 데이터의 형식은 **Device profile**에서 기술됩니다.

CIP 디바이스는 *I/O* 클라이언트와 *I/O* 서버가 동시에 될 수 있습니다. 하나의 디바이스에서 다중 *I/O* 연결들도 가능합니다.

메시지를 전송하기 위한 내재된 계층들에 대한 **CIP I/O** 메시지의 적용은, 예를 들어 **CAN** 또는 **UDP/IP**를 사용하여, **CIP** 구현의 일부이며 **DeviceNet**, **ControlNet** 과 **EtherNet/IP**에 따라 달라집니다.

명시적 메시지 보내기의 예 (1)



명시적 메시징은 Request/Response 관계를 사용합니다. 여기 우리의 예에서 메리는 폴에게 데이터를 요청합니다. 그녀는 그의 키를 알고 싶어합니다. 이를 위해 그녀는 CIP 서비스 „Get_Attribute_Single“ 을 사용합니다. 폴은 이 예에서 1.8m 인 그의 속성 값 „height“ 를 성공적으로 응답합니다.

명시적 메시지 보내기의 예 (2)



당신의 립스틱은
무슨 색깔입니까?

나는 립스틱을
사용하지 않습니다!

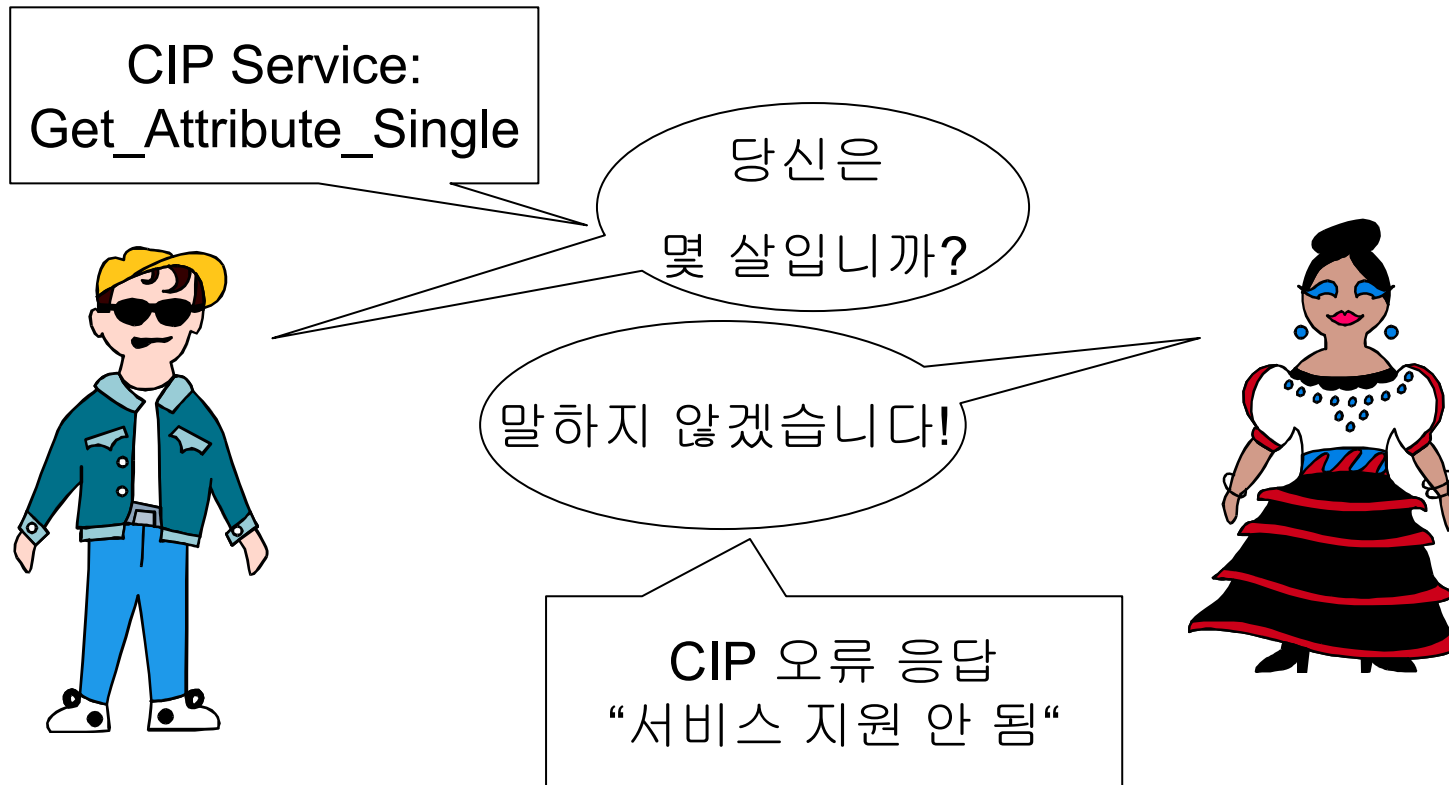
CIP 오류 응답
“속성 지원 안 됨“

CIP Service:
Get_Attribute_Single



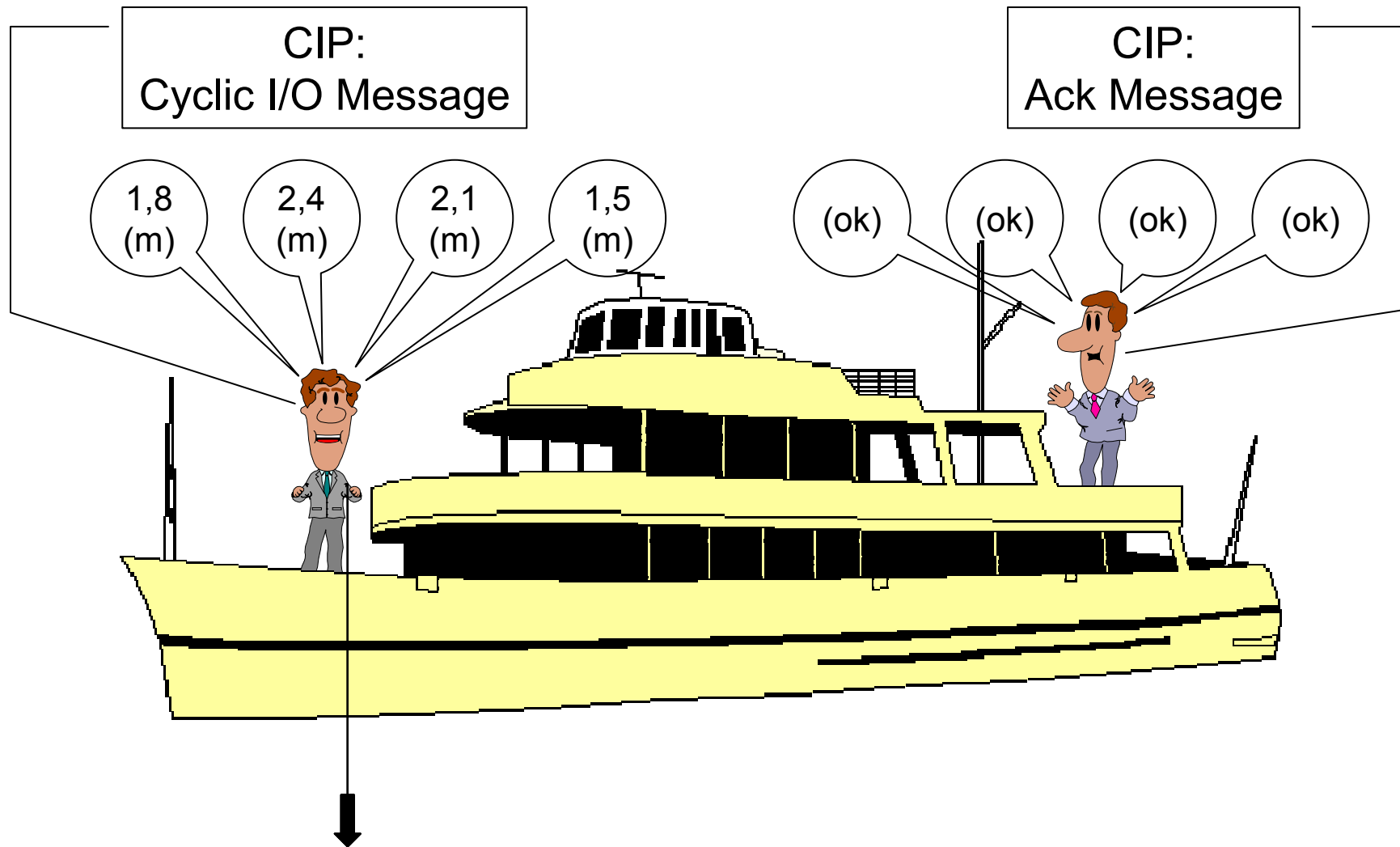
위의 예에서 메리는 폴에게 그가 립스틱을 사용하는지 묻습니다.
폴은 이 속성을 지원하지 않으므로 CIP 오류 응답 메시지 „Attribute not supported“ 로 응답합니다.

명시적 메시지 보내기의 예 (3)



명시적 메시징을 위한 마지막 예에서, 폴은 메리의 나이를 알고 싶어합니다. 메리는 이 속성을 지원하지만 폴에게 그 값을 말하려 하지 않습니다. 그러므로 그녀는 CIP 오류 응답 메시지 „Service not supported“로 응답합니다. 이것은 명시적 메시징의 매우 간단한 예였습니다. CIP 공통 규격에서는 더욱 많은 CIP 서비스들이 정의됩니다. CIP 서비스들이 디바이스에서 데이터를 읽고 쓰는데만 사용될 수 없으며, 그들은 더욱 강력합니다, 왜냐하면 사용자는 그것들을 사용하여 (Reset-Service 처럼) 디바이스의 내부 상태를 변경할 수 있기 때문입니다. 또한 I/O 연결의 설립과 구성은 명시적 메시징 연결을 거쳐 CIP 서비스를 사용하여 관리됩니다.

I/O 메시지 보내기의 예



I/O 메시지 보내기의 예 (보충설명)

CIP에서 정의된 두 번째 메시지 유형은 I/O 메시지입니다. 다음 예에서 우리는 I/O 연결을 위한 두 개의 말단 (point-to-point 연결)을 가정합니다:

선장 (그는 네트워크에서 클라이언트와 창시자가 될 수 있다고 가정)과 조타수는 대양 항로의 깊이를 측정합니다. 선장과 조타수와의 관계가 설정된 후에, 조타수는 주기적으로 깊이를 측정하고 선장에게 결과를 소리칩니다. 선장 자신은 그 데이터를 받았는지 확인합니다.

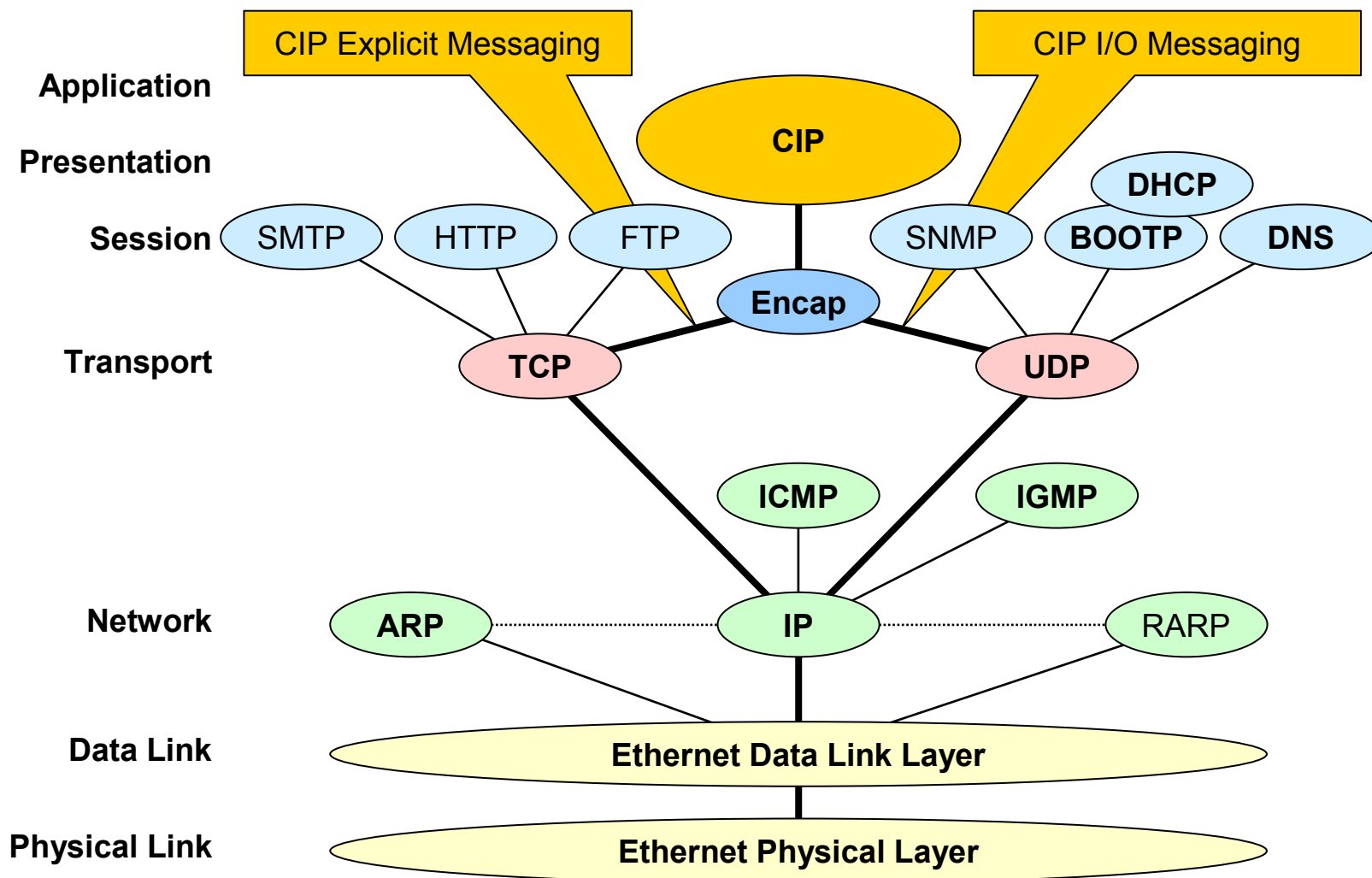
명시적 메시징과 비교했을때 큰 차이점은:

- 양 쪽 말단들이 데이터의 포맷과 데이터에 관한 지식을 알고 있어야 합니다. 여기 우리 예에서 깊이는 미터로 측정됩니다.
- 여기서는 어떤 CIP 서비스도 사용되지 않습니다. 따라서 어떤 프로토콜 오버헤드도 없으며, 단지 raw I/O 데이터가 네트워크를 통해 전송됩니다.

EtherNet/IP 기본 원리

Ethernet & TCP/IP 입문

중간 계층(Middle Layers)



중간 계층 (보충설명)

EtherNet/IP의 주요 혜택 중의 하나는 CIP 메시징을 위한 일반 TCP/IP 프로토콜의 사용입니다. EtherNet/IP 는 웹서버 (HTTP 사용) 또는 mail-alerter (SMTP 사용) 같은 디바이스에서 다른 프로토콜들과 병렬로 실행될 수 있습니다.

CIP *Explicit Messaging* 은 *TCP/IP* 네트워킹을 사용하는 반면 CIP *I/O Messaging*은 *UDP/IP* 네트워킹을 사용합니다. 이것들 모두 EtherNet/IP 규격의 일부인 일반 *Encapsulation Protocol* 을 사용합니다.

Ethernet (1)

EtherNet/IP은 Data Link 계층과 물리 계층으로서 IEEE 802.3 (Ethernet) 을 사용합니다

Star 위상 / Tree 위상

- 10 Base-T (10 Mbps)
- 100 Base-T (100 Mbps)

충돌 검사를 사용하는 반송파 감지 다중 액세스 (CSMA/CD)

- 충돌을 피하기 위해 스위치가 권장됩니다

Ethernet (1) (보충설명)

EtherNet/IP 는 IEEE 802.3 표준에 따라 이더넷을 사용합니다.

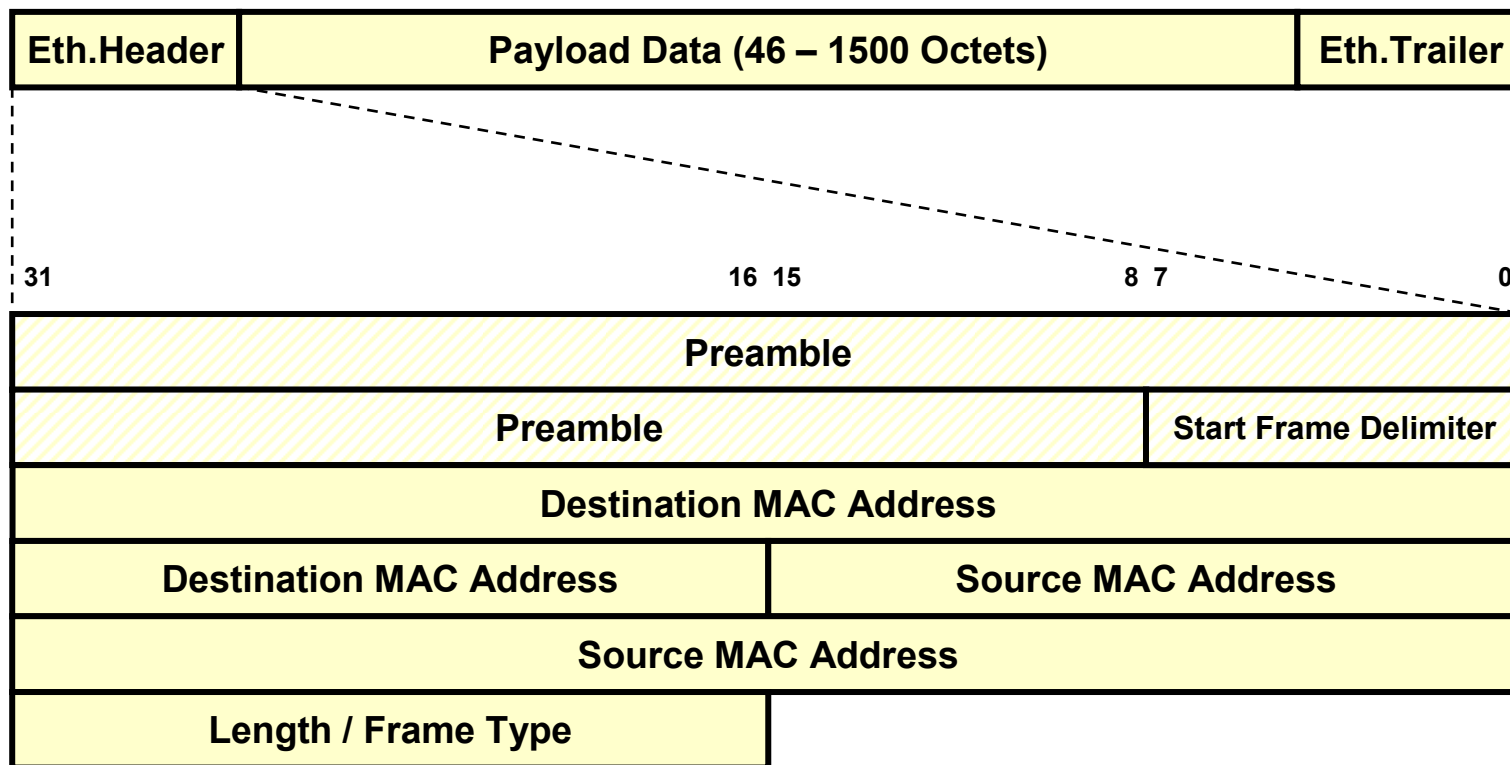
주요 특성들 중의 하나는 네트워크에서 허브나 스위치 같은 기반 요소들을 필요로 하는 *Star* 또는 *Tree Topology* 의 사용입니다. 예를 들어 DeviceNet과 구분하는 편리한 *Line Topology* 는 EtherNet/IP 네트워크에서 사용될 수 없습니다.

EtherNet/IP에서 사용되는 물리 계층에 대한 두 가지 표준은 10 Base-T 와 100 Base-T 입니다.

이더넷에서 물리 계층에 연결된 디바이스의 동시 액세스는 충돌을 야기합니다. 이러한 것들은 이더넷 컨트롤러에 의해 해결됩니다. 충돌은 네트워크에서 메시지의 지연을 증가시키며 따라서 네트워크의 실시간 행동과 결정력을 저하시킵니다.

Ethernet (2)

EtherNet/IP 는 Data Link 계층과 물리 계층으로서 IEEE 802.3 (Ethernet) 을 사용합니다



Ethernet (2) (보충설명)

모든 EtherNet/IP 디바이스들 최소 지원:

- RFC 894 에서 정의된 대로 IEEE 802.3 (Ethernet)

IEEE 802.3 이더넷 프레임은 Header, Payload Data 와 Trailer 로 구성됩니다.

이더넷 헤더는 수신 노드들의 동기화를 돕기 위해 Preamble 과 Start Frame Delimiter 와 같이 시작합니다 (이러한 8 octets 들은 우리가 프레임 길이를 얘기할 때 대개 제외됩니다).

헤더의 다른 요소는 Destination MAC Address, Source MAC Address 와 Length/Frame Type 입니다.

이더넷 트레일러는 4 옥테트의 Frame Check Sequence 를 포함합니다.

IEEE 802.3 이더넷 프레임의 최소 길이는 64 옥테트입니다. 실제 payload 데이터가 46 옥테트 미만이라면, 46 옥테트를 얻기 위해 0으로 된 패딩 바이트가 채워집니다.

IEEE 802.3 이더넷 프레임의 최대 길이는 1518 옥테트입니다.

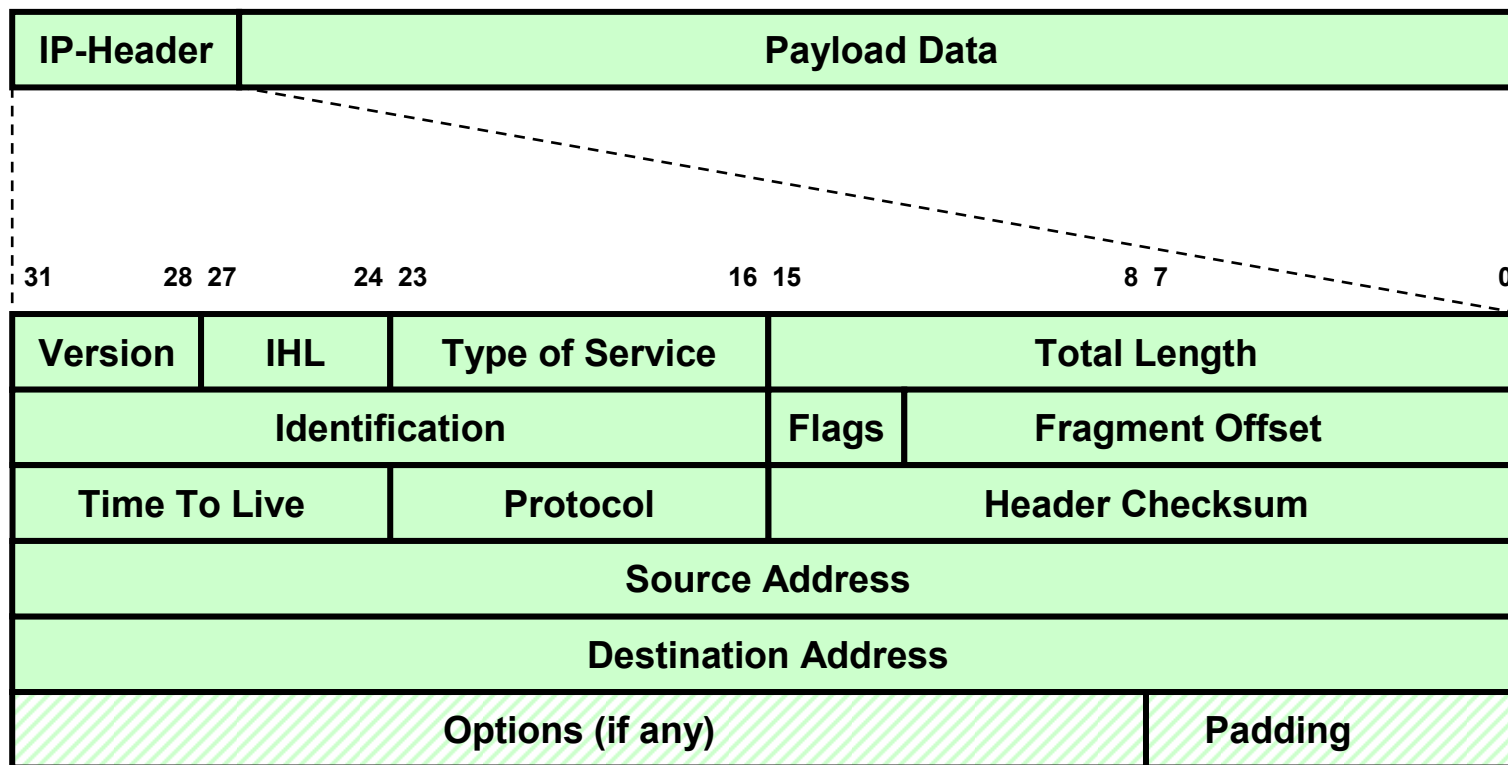
프로토콜 오버헤드가 없다고 가정하면 100 Mbps의 네트워크에서 이론적인 데이터 처리량은 12.5 MByte/s 입니다.

충돌이 없다고 가정하면 100 Mbps의 IEEE 802.3 네트워크에서 이론적 순수 데이터 처리량은 다음과 같습니다;

- 1 byte payload data (+ 45 padding bytes) → 164 kByte/s
- 46 byte payload data (minimum IEEE 802.3 frame) → 7.5 MByte/s
- 1500 byte payload data (maximum IEEE 802.3 frame) → 12.2 MByte/s

인터넷 프로토콜 (IP)

EtherNet/IP 는 인터넷 프로토콜(IPv4)을 사용합니다



인터넷 프로토콜 (IP) (보충설명)

모든 EtherNet/IP 디바이스 최소 지원:

- Internet Protocol (IP version 4) (RFC 791)

IP 프레임은 Header 와 Payload Data로 구성됩니다. 최소 헤더 길이는 20 octets 입니다. 사용되는 옵션에 따라, 그 크기는 최대 24 octets가 될 수 있습니다.

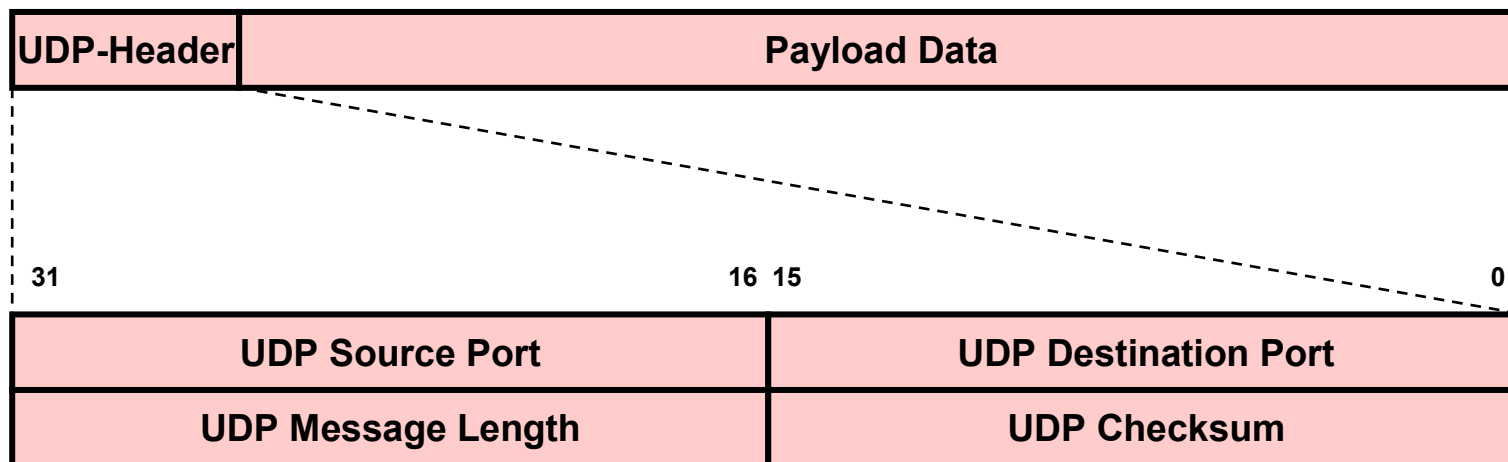
헤더의 몇 가지 중요 요소들은 다음과 같습니다:

- 전체 길이
 - payload 데이터를 위한 최대 크기는 65535 octets 입니다
- Source Address 와 Destination Address
 - 192.168.1.100 와 같은 IP 주소가 있습니다

IP 프로토콜은 단편화(fragmentation)를 지원합니다. 따라서 최대 65535 octets의 payload 데이터가 이더넷과 같은 하위 계층을 통해 전송될 수 있으며, 이때 payload 데이터는 1500 octets로 제한됩니다.

사용자 데이터그램 프로토콜 (UDP)

Datagram 기반, 비-신뢰, 비연결형 패킷 전달 서비스



EtherNet/IP 은 I/O 메시지 전송에 UDP를 사용합니다

사용자 데이터그램 프로토콜 (보충설명)

모든 EtherNet/IP 디바이스 최소 지원:

- User Datagram Protocol (UDP) (RFC 768)

UDP 프레임은 Header 와 Payload Data로 구성됩니다. 이 헤더는 항상 8 octets를 포함합니다. 헤더와 UDP를 이용해 전송된 payload 데이터의 크기는 65535 octets로 제한됩니다. 그러면 순수 payload 데이터는 65527 octets가 됩니다.

이 계층에서는 단편화 프로토콜이 없습니다.

User Datagram Protocol 은 *Ports* 를 사용하여 노드에서 실행하는 다양한 애플리케이션들에 대한 datagrams을 demultiplex 합니다.

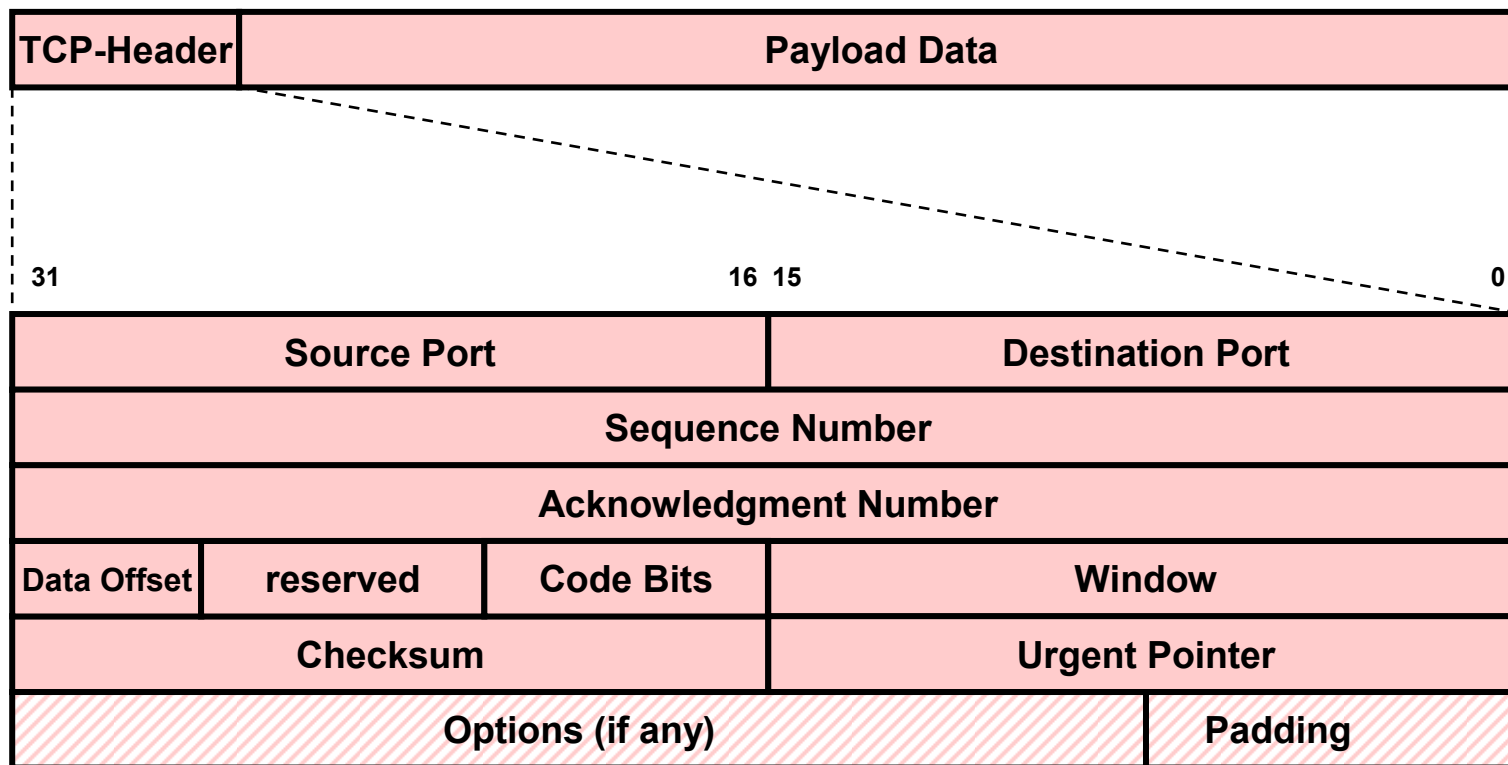
몇 가지 „잘 알려진 포트들“의 예 입니다.

- 53 Domain Name Server (DNS Server)
67 Bootstrap Protocol (BOOTP Server)
68 Bootstrap Protocol (BOOTP Client)
161 Simple Network Management Protocol (SNMP)

EtherNet/IP의 경우 UDP-Port 2222 (0x08AE) 가 CIP I/O Messaging 에 사용됩니다.

전송 제어 프로토콜 (TCP)

Stream 기반, 신뢰성 있는 연결 지향 프로토콜



EtherNet/IP 는 명시적 메시지 전송에 TCP를 사용합니다

전송 제어 프로토콜 (보충설명)

모든 EtherNet/IP 디바이스 최소 지원:

- Transmission Control Protocol (TCP) (RFC 793)

TCP 세그먼트는 **Header**와 **Payload Data**로 구성됩니다. 최소 헤더 길이는 **20 octets**입니다. 사용된 옵션에 따라, 그 크기는 최대 **24 octets**가 될 수 있습니다.

TCP 를 사용하여 전송된 **payload** 데이터의 크기는 무제한입니다 – 이것은 우리가 **Stream** 으로 알고 있는 것입니다.

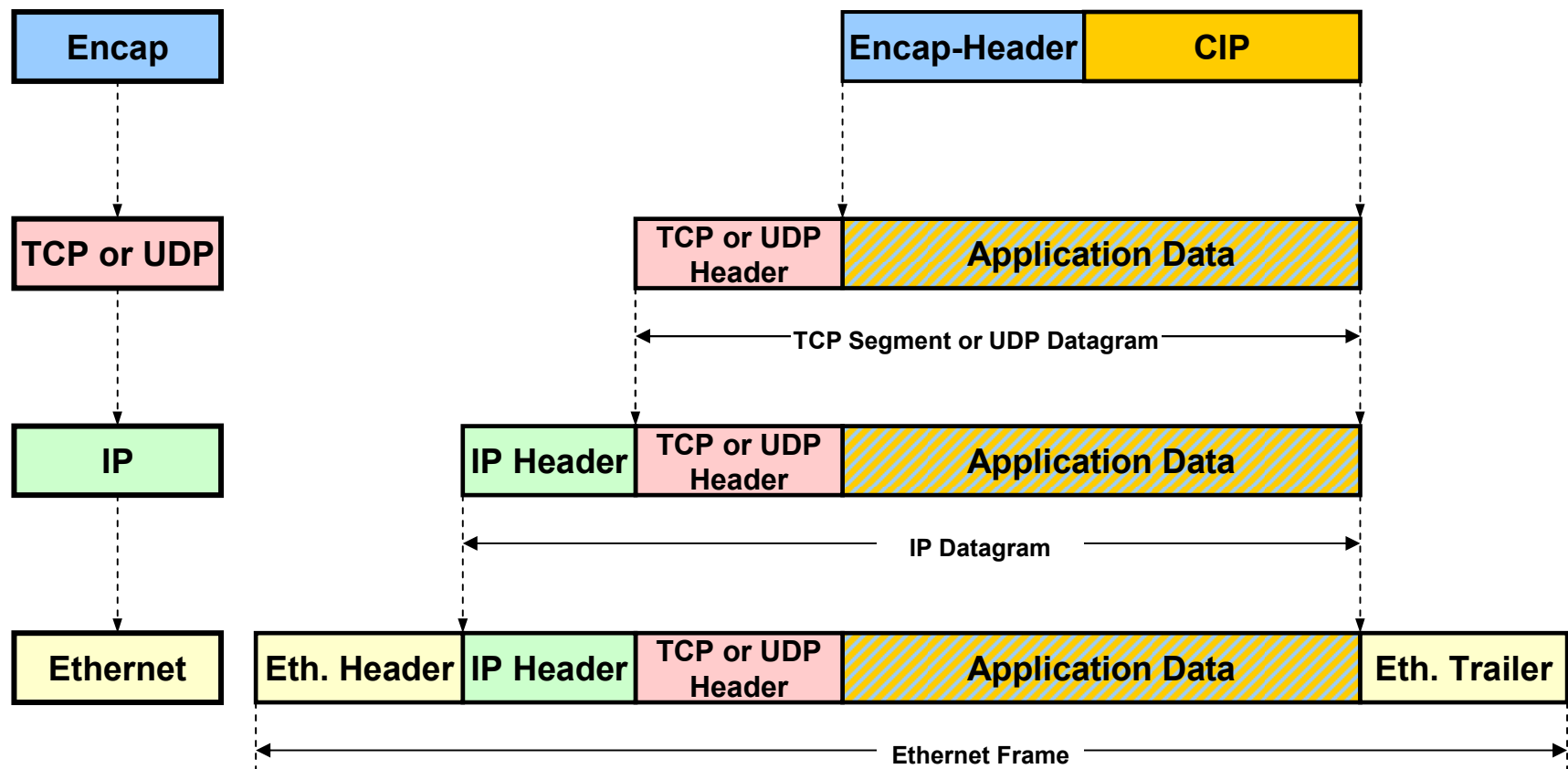
TCP 또한 **Ports**를 사용합니다, 몇 가지 „잘 알려진 포트들“ 의 예입니다.

- 20 File Transfer Protocol (FTP Data)
- 21 File Transfer Protocol (FTP Control)
- 23 Telnet
- 25 Simple Mail Transfer Protocol (SMTP)
- 80 Webserver (HTTP)

EtherNet/IP은 TCP-Port 44818(0xAF12) 가 Encapsulation 프로토콜과 이를 통한 CIP Explicit Messaging 에 사용됩니다

종합 ...

... 이것이 우리가 „캡슐화(Encapsulation)“로 알고 있는 것입니다



종합 ... (보충설명)

프로토콜들은 하나에서 다른 것으로 캡슐화됩니다.

IP Datagram은 이더넷 프레임의 payload data에서 캡슐화됩니다.

TCP Segments 또는 UDP Datagrams 은 IP Datagrams의 payload data에서 캡슐화됩니다.

끝으로, CIP 프로토콜은 캡슐화 프로토콜을 사용하여 TCP Segment /UDP Datagram의 payload data에서 캡슐화됩니다. 우리는 이 교육의 후반부에서 CIP Encapsulation 프로토콜에 대해 더 자세히 배울 것입니다.

이 그림은 스택(stack) 같아 보이며, 그것이 TCP/IP 프로토콜 소프트웨어가 또한 TCP/IP 프로토콜 스택으로 알려진 이유입니다.

그 외 요구되는 프로토콜들

ARP – Address Resolution Protocol

ICMP – Internet Control Messaging Protocol

IGMP – Internet Group Management Protocol

그 외 요구되는 프로토콜들 (보충설명)

모든 EtherNet/IP 디바이스 최소 지원:

- Address Resolution Protocol (ARP) (RFC 826)
 - ARP는 이더넷 계층에서 IP 계층에서부터 MAC 주소까지 IP 주소들을 전송합니다.
- Internet Control Messaging Protocol (ICMP) (RFC 792)
 - ICMP는 IP 프로토콜의 필수 부분이며 예를 들어 데이터그램 처리에서 오류를 보고할 때 사용됩니다.
- Internet Group Management Protocol (IGMP) (RFC 1112 & 2236)
 - IGMP 또한 IP Multicasting 을 관리하기 위한 IP 프로토콜의 필수 부분입니다. EtherNet/IP의 경우 이 프로토콜은 매우 중요합니다, 왜냐하면 I/O 메시지들이 IP Multicast 를 이용할 수 있습니다.
 - 주의: 비록 IGMP가 IP의 필수 부분이라 하더라도 모든 TCP/IP 프로토콜 스택들에 의해 제공되지는 않습니다. 여러분이 선택한 TCP/IP 프로토콜 스택이 IGMP를 지원하는지 확인하시기 바랍니다!

선택 프로토콜들

BOOTP – Bootstrap Protocol

DHCP – Dynamic Host Configuration Protocol

DNS – Domain Name System

선택 프로토콜들 (보충설명)

EtherNet/IP 디바이스들은 다음의 프로토콜들을 지원합니다 :

- Bootstrap Protocol (BOOTP) (RFC 951)
 - BOOTP는 IP 주소를 포함하여 서버로부터 startup 정보를 얻기 위해 호스트에 의해 사용됩니다. 이 프로토콜은 boot-up 시간 동안에 디바이스로 유효한 IP 주소를 배정하기 위해 Ethernet/IP 네트워크에서 사용되는 한 가지 수단입니다. 또 다른 방법은
- Dynamic Host Configuration Protocol (DHCP) (RFC2131)
 - DHCP는 BOOTP (RFC 1534 참고) 에 바탕을 두며 이 정보를 더욱 편리하게 얻을 수 있도록 개선되었습니다.
- Domain Name System (DNS) (RFC 1034 and 1035)
 - DNS 서버는 DNS 클라이언트들을 위해 컴퓨터 이름을 IP 주소로 변환하는데 사용됩니다.

EtherNet/IP 기본 원리

이더넷에 대한 CIP의 적용으로서의 EtherNet/IP

EtherNet/IP – 용어

클라이언트/서버 – application view

- *Client* 는 *Server*에 메시지 요청을 개시합니다

디바이스 기능의 네 가지 분류

- 메시지 서버
 - 이것은 **CIP** 디바이스의 최소 요건입니다
- I/O 서버와 메시지 서버
 - better known as **Adapter Class**
- 메시지 서버와 메시지 클라이언트
- 메시지 서버와 메시지 클라이언트, I/O 서버와 I/O 클라이언트
 - **Scanner Class** 로 더 잘 알려짐

EtherNet/IP – 용어 (보충설명)

애플리케이션의 관점에서 우리는 클라이언트와 서버를 구분합니다. 클라이언트는 서버에 메시지를 요청하는 연결의 종점(endpoint)입니다

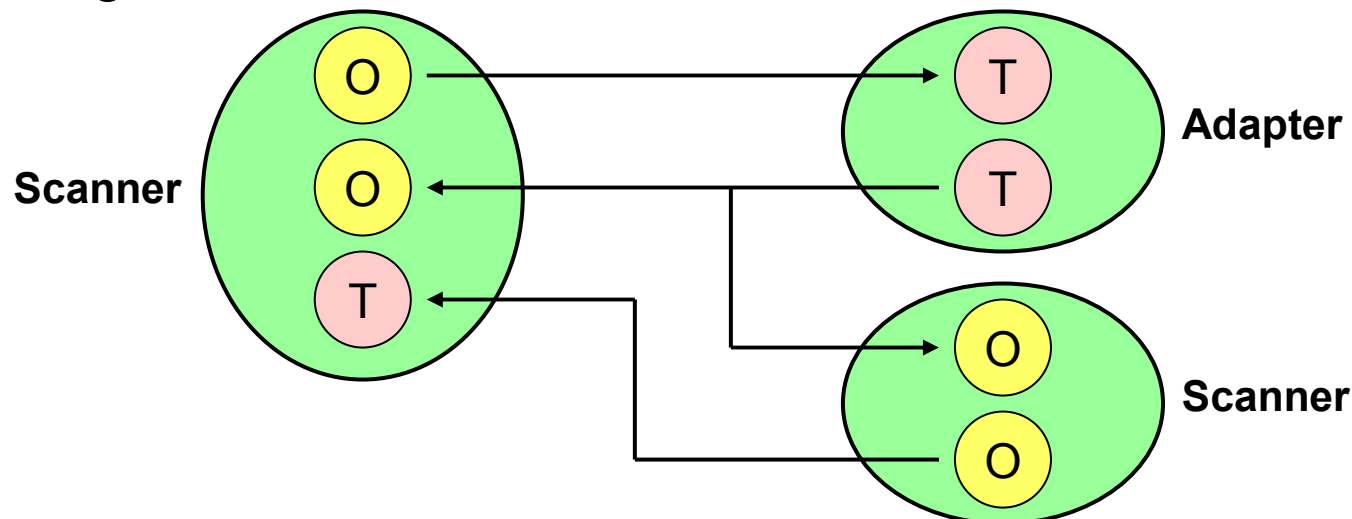
EtherNet/IP의 경우 우리는 이제 4가지 디바이스 레벨을 정의할 수 있습니다 – 그것들이 클라이언트 또는 서버인지 그리고 그들이 Explicit Messaging 또는 I/O Messaging을 지원하는가에 따라.

EtherNet/IP에서는 대개 다음의 생략형을 사용합니다: „Message Server“ 용어는 „Explicit Messaging Server“ 에 대한 단축 형태이며 „I/O Server“ 용어는 „I/O Messaging Server“에 대한 단축 형태입니다.

EtherNet/IP – 용어

Originator/Target – network view

- *Originator* 는 *Target*에 대한 연결을 개시합니다
 - O->T: originator는 타겟에 의해 소비되는 데이터를 만듭니다
 - T->O: 타겟은 originator에 의해 소비되는 데이터를 만듭니다
- 한 디바이스가 동시에 서로 다른 연결에 대해 originator와 target 이 될 수 있습니다



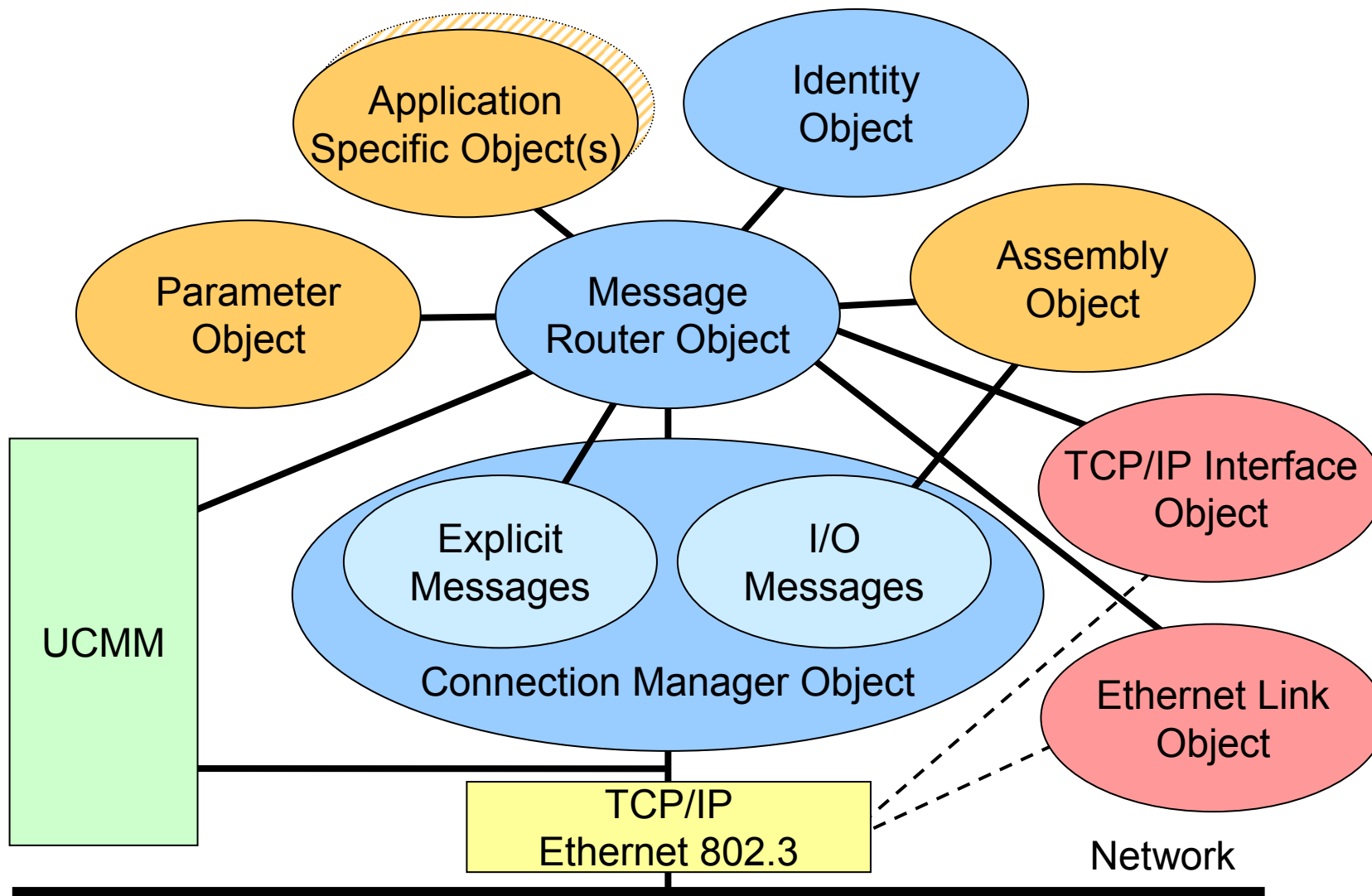
EtherNet/IP – 용어 (보충설명)

우리가 네트워크 관점에서 볼 때, 한 디바이스는 다른 디바이스들에 대해 열려진 여러 개의 연결들을 가질 수 있습니다. 각 연결은 **originator**, 이것은 연결을 개시하는 디바이스 입니다, 그리고 한 개 또는 그 이상의 타겟들, 연결의 말단(endpoints)를 갖고 있어야 합니다.

EtherNet/IP 에서 데이터는 일반적으로 **Originator** 에서 **Target**으로 그리고 **Target**에서 **Originator**로 양쪽 방향으로 교환됩니다.

O -> T 와 T -> O 용어가 여기서는 일반적인 약자입니다.

객체 모델



객체 모델 (보충설명)

적어도 다음 오브젝트들 중의 한 가지 **instance** 는 디바이스에서 구현되어야 합니다

CIP 공통 오브젝트

- Identity Object (Class Code 0x01)
Message Router Object (Class Code 0x02)
Connection Manager Object (Class Code 0x06)

EtherNet/IP 특정 오브젝트

- TCP/IP Interface Object (Class Code 0xF5)
Ethernet Link Object (Class Code 0xF6)

Unconnected Message Manager (UCMM)

EtherNet/IP 디바이스는 CIP common 오브젝트 라이브러리를 공유합니다
Reference: Volume 1: CIP Specification, Chapter 5 – Object Library

EtherNet/IP 디바이스는 CIP common device profiles을 공유합니다
Reference: Volume 1: CIP Specification, Chapter 6 – Device Profiles

TCP/IP 인터페이스 오브젝트

- TCP/IP 인터페이스 오브젝트는 디바이스의 TCP/IP 네트워크 인터페이스를 구성하는 메커니즘을 제공합니다
- 각각의 **instance** 는 TCP/IP 네트워크 인터페이스를 구성하기 위한 6개의 **instance** 속성들을 제공합니다
 - Status
 - Configuration Capability
 - Configuration Control
 - Physical Link Object
 - Interface Configuration
 - Host Name

TCP/IP 인터페이스 오브젝트 (보충설명)

EtherNet/IP의 경우 디바이스는 최소한으로 이 오브젝트의 하나의 instance를 지원할 것입니다.

TCP/IP Interface Object의 각 instance 는 정확히 하나의 TCP/IP 네트워크 인터페이스에 일치합니다.

Scanners와 Adapters는 대개 이 오브젝트의 한 instance를 지원합니다. Bridges, Gateways 와 CIP Routers는 이 오브젝트의 두 개 또는 그 이상의 instances를 지원할 수 있습니다.

TCP/IP Interface Object instance 속성들은 디바이스에서 내재된 TCP/IP 프로토콜 스택의 데이터 구조에 연결됩니다.

Ethernet 연결 오브젝트

- Ethernet Link Object 는 Ethernet 802.3 통신 인터페이스를 위한 link 특수 카운터와 상태 정보를 보유합니다
- Ethernet Link Object의 각 instance는 하나의 Ethernet 802.3 통신 인터페이스에 정확히 부합합니다
 - Interface Speed
 - Interface Flags
 - Physical Address (MAC)
 - Interface Counters
 - Media Counters
 - Interface Control

Ethernet 연결 오브젝트 (보충설명)

EtherNet/IP의 경우 Ethernet 802.3이 사용된다고 가정할 때 최소한 이 오브젝트의 instance 한 개가 지원될 것입니다.

Ethernet Link Object의 각 instance는 정확히 하나의 Ethernet 802.3 통신 인터페이스에 일치합니다.

Scanners와 Adapters는 대개 이 오브젝트의 instance 하나를 지원합니다. Bridges, Gateways 와 CIP Routers는 이 오브젝트의 두 개 또는 그 이상의 instances를 지원할 수 있습니다.

Ethernet Link Object instance 속성들은 디바이스에서 내재된 Ethernet 드라이버의 데이터 구조에 연결됩니다. 이것은 이더넷 드라이버가 이러한 데이터를 제공하고 그 데이터가 CIP 서비스를 이용하여 정확히 액세스되는지를 확인하기 위한 이 오브젝트 구현의 일부입니다.

물리 계층 - Indicators

Indicators 는 선택사항입니다

Indicator가 제품에서 구현되면, 이것은 규격에 충실해야 합니다

목표: 모든 3개 CIP 기술들의 모습과 느낌 공통: DeviceNet, ControlNet 과 EtherNet/IP

- 2색의 빨강/초록 indicator (일반적인 LED)
- Power-Up 동안의 자체-테스트
- Flash 속도 초당 1, Duty Cycle 50 %

물리 계층 – Indicators (보충설명)

여러분이 다른 CIP 기반 기술들 DeviceNet 과 ControlNet에 익숙해지면 indicator 에 대해 알게 됩니다. 이것들은 네트워크에서 결함 디바이스들을 찾아내기 위한 서비스와 유지보수 담당자를 도와줍니다. 이를 위해 전원 공급시 indicator가 작동되는 것을 보여주기 위한 셀프 테스트는 필수입니다.

이것은 디바이스의 내부 상태를 보여주고 오류가 발생한 경우 그것들을 보여줄 수 있는 가능한 간단한 indicators의 coding을 유지하기 위함 이였습니다.

Indicators의 구현이 해석(interpretation)을 위한 여지를 남겨두지 않는다는 것이 중요합니다. Indicator가 구현될 때 이것은 엄격하게 규격에 충실합니다. Indicators의 정확한 작동을 확인하는 것은 ODVA Conformance Test의 일부입니다. DeviceNet에서 이것은 물리 계층 테스트를 통해 실행되며, EtherNet/IP의 경우 앞으로 그 절차가 정의되어야 합니다.

규격을 준수하는 모든 indicators는 디바이스의 하우징의 어떤 부분도 제거하지 않고 볼 수 있습니다. 또한 Indicator의 뜻을 명확하게 보여주는 label 도 있습니다.

물리 계층 – Performance Levels

EtherNet/IP 는 두 가지 레벨을 지정합니다

- 판매용 기성품 (Commercial Off The Shelf; COTS)
 - 사무용 네트워크로 유명
 - ANSI/TIA/EIA-568 B.2 standard
 - IEEE 802.3u / TP-PMD standard
 - COTS 컴포넌트들은 사용자 시스템의 성능을 저하시킬 수 있습니다
- 산업용
 - COTS와 비교했을 때 필수 요건 증가
- 두 개의 레벨들은 다음을 위한 요건들을 서로 다르게 정의합니다
 - Cabling 과 Connectors
 - 위상 제약조건

물리 계층 – Performance Levels (보충설명)

EtherNet/IP 규격은 디바이스들의 두 가지 *Performance Levels* 을 정의합니다: *Commercial of the Shelf* 와 *Industrial*.

Commercial off the Shelf (COTS) 레벨은 온도 또는 밀봉(sealing)에 대한 요건이 매우 낮거나 지정되지 않은 사무 환경을 위해 정의된 저 표준을 사용합니다.

Industrial 레벨은 매우 높은 표준을 설정하는데, 온도, 밀봉, voltage isolation, 등에 관한 강화된 요건들이 정의됩니다. EtherNet/IP의 제품들은 *Industrial* 표준을 따를 것을 권장합니다.

EtherNet/IP 기본 원리

적합성 검사(Conformance Testing)

적합성 검사

EtherNet/IP 제품들은 반드시 적합성이 검사되어야 합니다

- ODVA Protocol Conformance Test Software

ODVA 검사 서비스 제공

- ODVA Training and Technology Center
앤 아버, 미시건 – 미국
- CVS@IAF
Otto-von-Guericke-University
마그데부르크, 독일