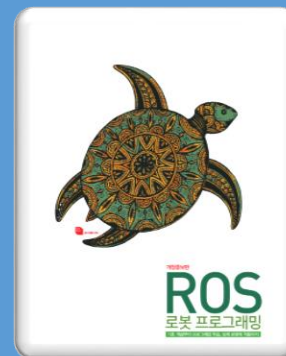


# ROS의 중요 컨셉

**ROBOTIS**

Open Source Team

Yoonseok Pyo

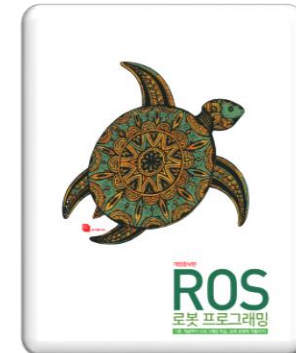


교재  
P. 42~135

# Contents

---

- I. ROS 용어 정리
- II. 메시지 통신
- III. 메시지
- IV. 네임
- V. 좌표 변환(TF)
- VI. 클라이언트 라이브러리
- VII. 이기종 디바이스 간의 통신
- VIII. 파일 시스템
- IX. 빌드 시스템



교재  
P. 42~135

로봇 운영체제 ROS 용어!

# ROS 용어

---

## ■ Node

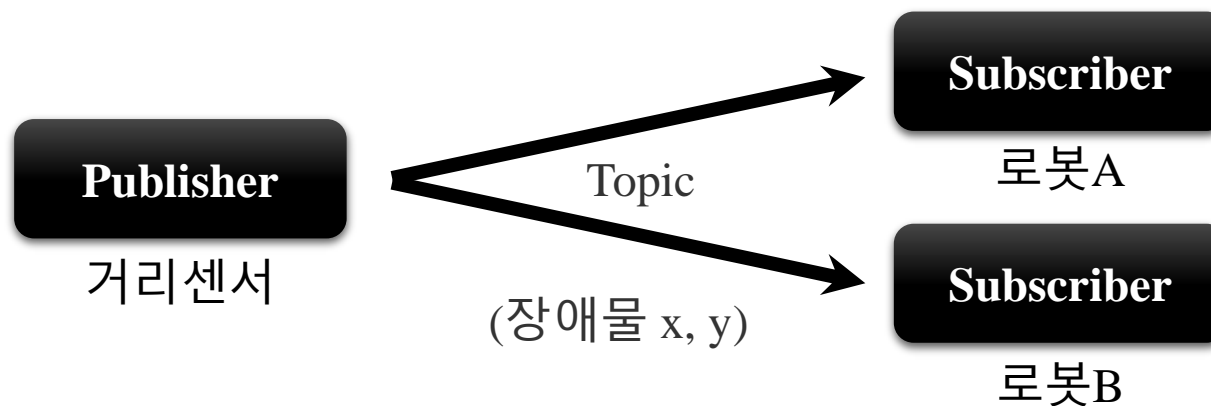
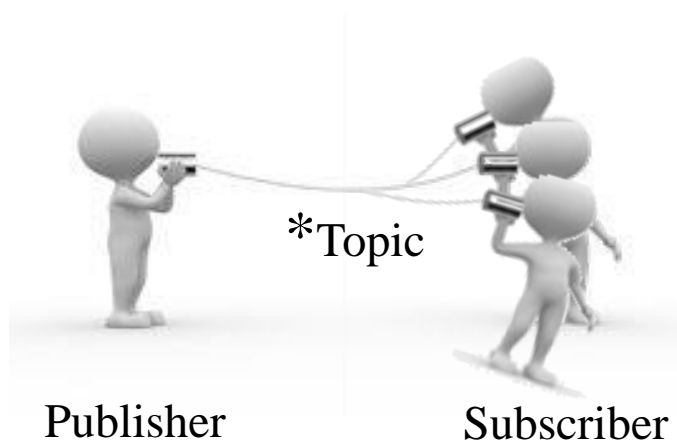
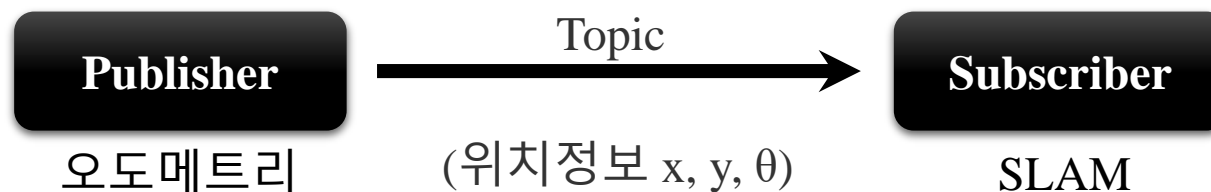
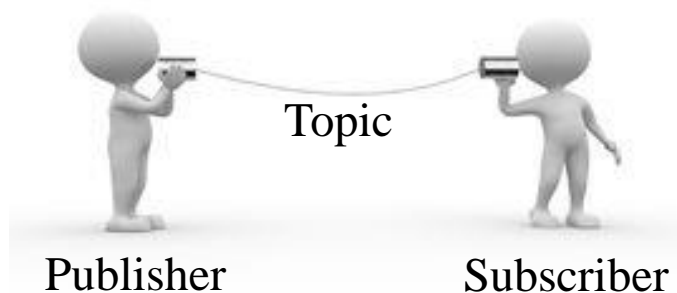
- 최소 단위의 실행 가능한 프로세서를 가리키는 용어으로써 하나의 실행 가능한 프로그램으로 생각하면 된다. ROS에서는 최소한의 실행단위로 프로그램을 나누어 작업하게 된다. 각 노드는 메시지 통신으로 데이터를 주고 받는다.

## ■ Package

- 하나 이상의 노드, 노드 실행을 위한 정보 등을 묶어 놓은 것. 또한, 패키지의 묶음을 메타패키지라 하여 따로 분리한다.

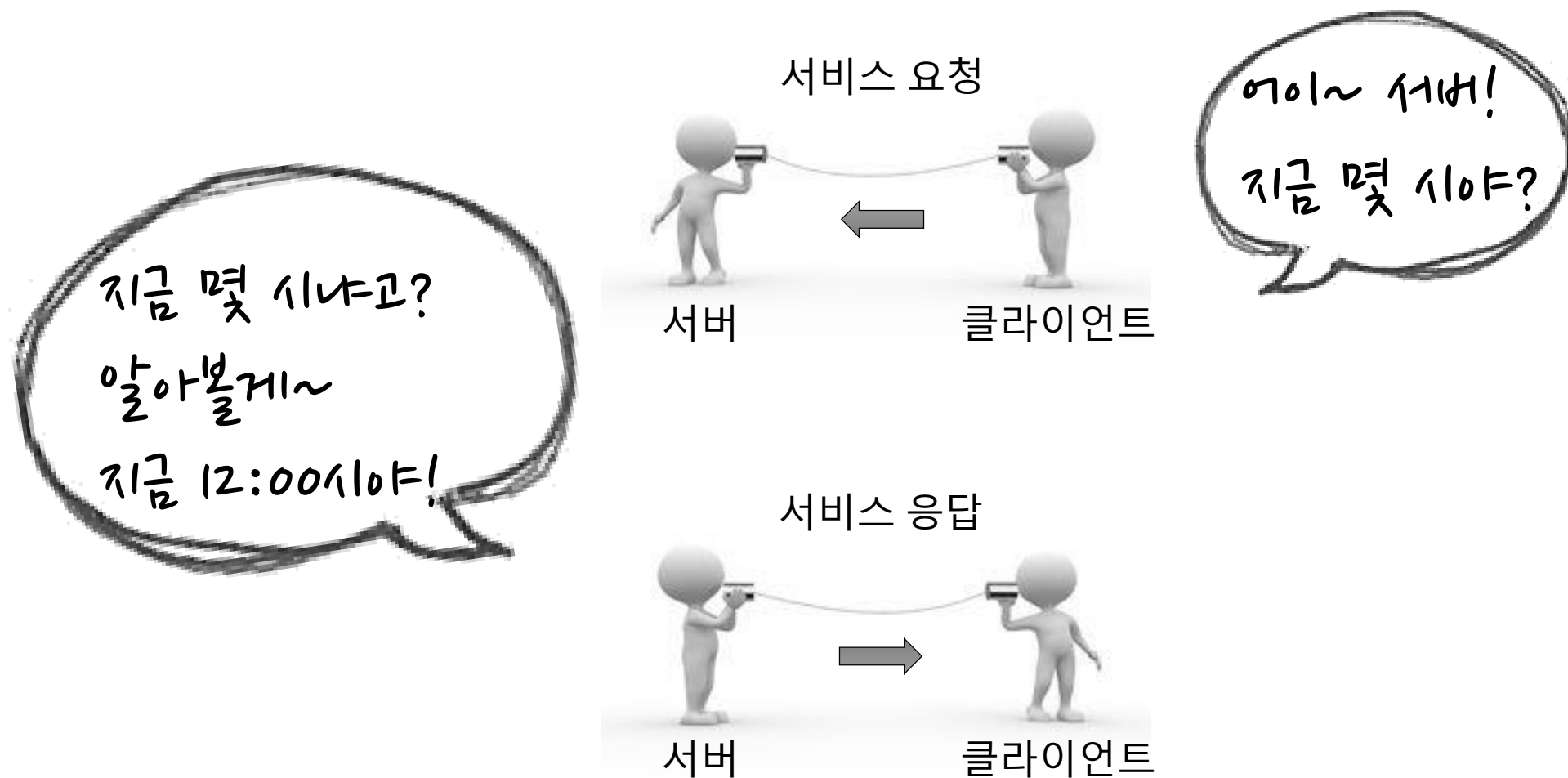
## ■ Message

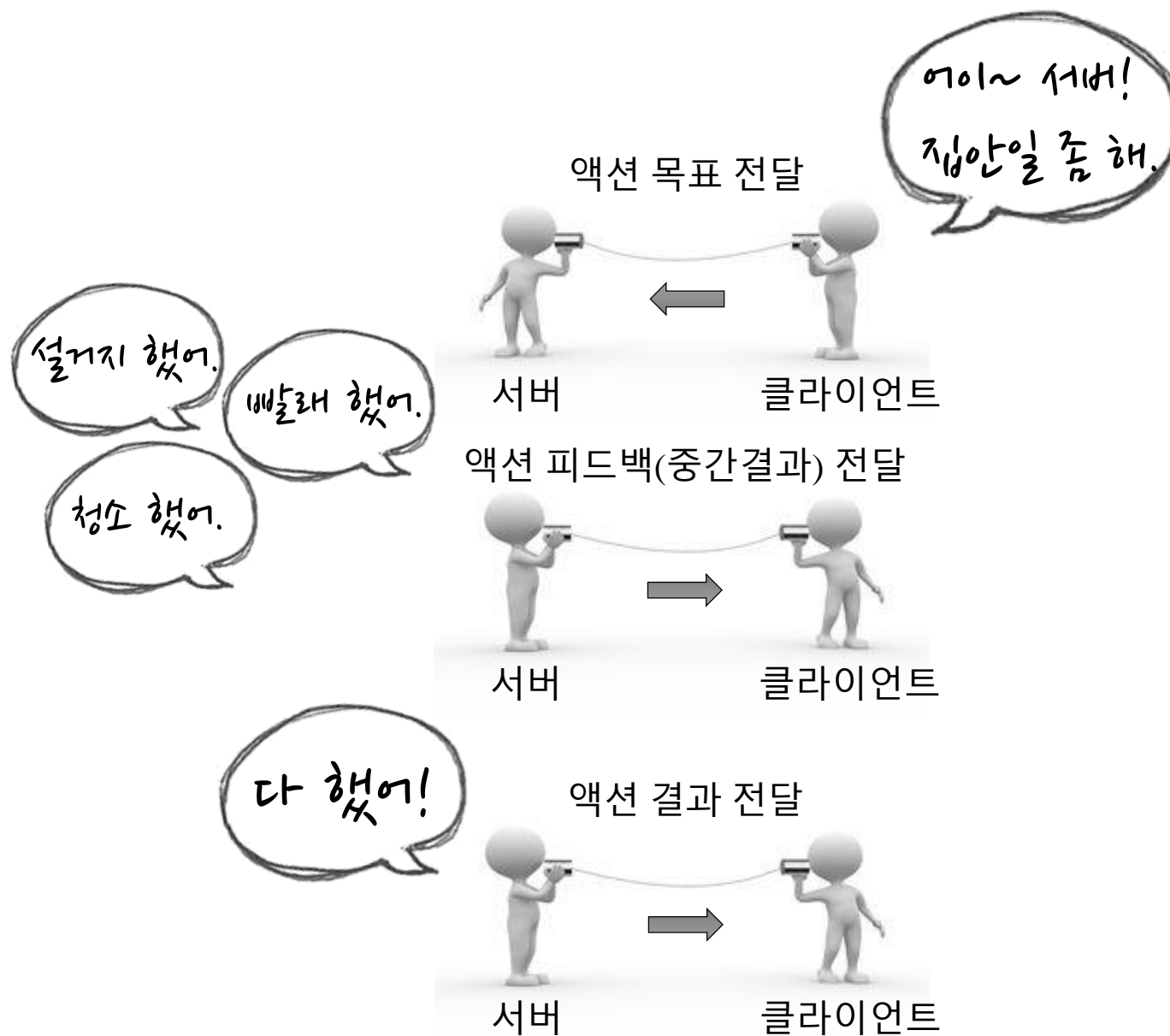
- 메시지를 통해 노드간의 데이터를 주고받게 된다. 메시지는 integer, floating point, boolean 와 같은 변수형태이다. 또한, 메시지 안에 메시지를 품고 있는 간단한 데이터 구조 및 메시지들의 배열과 같은 구조도 사용할 수 있다.



\*Topic 에 대해 1:1의 Publisher, Subscriber 통신도 가능하며, 목적에 따라서 1:N, N:1, N:N 통신도 가능하다.

# ROS 용어 Service, Service server, Service client





참 쉬죠?

이제 좀 더 자세히 알아보게요~ 😊

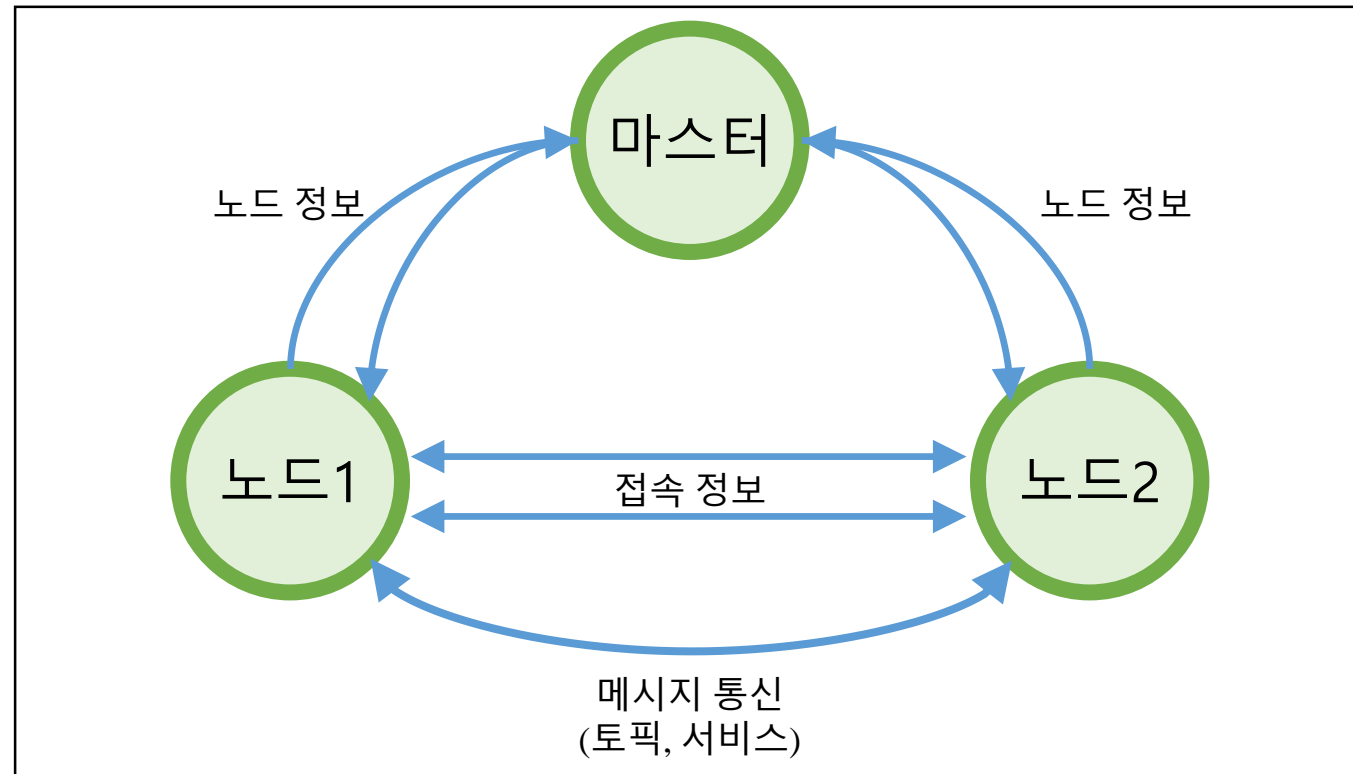


그 외 용도들은 강의 도중에 필요하면  
그때 그때 설명하겠습니다!

# 메시지 통신

# 메시지 통신 개념 잡기!

- ROS에서 가장 기본이 되는 기술적 포인트: 노드간의 메시지 통신!



# 메시지 통신 개념 잡기!

---

## 1. 마스터 구동: XMLRPC(XML-Remote Procedure Call)

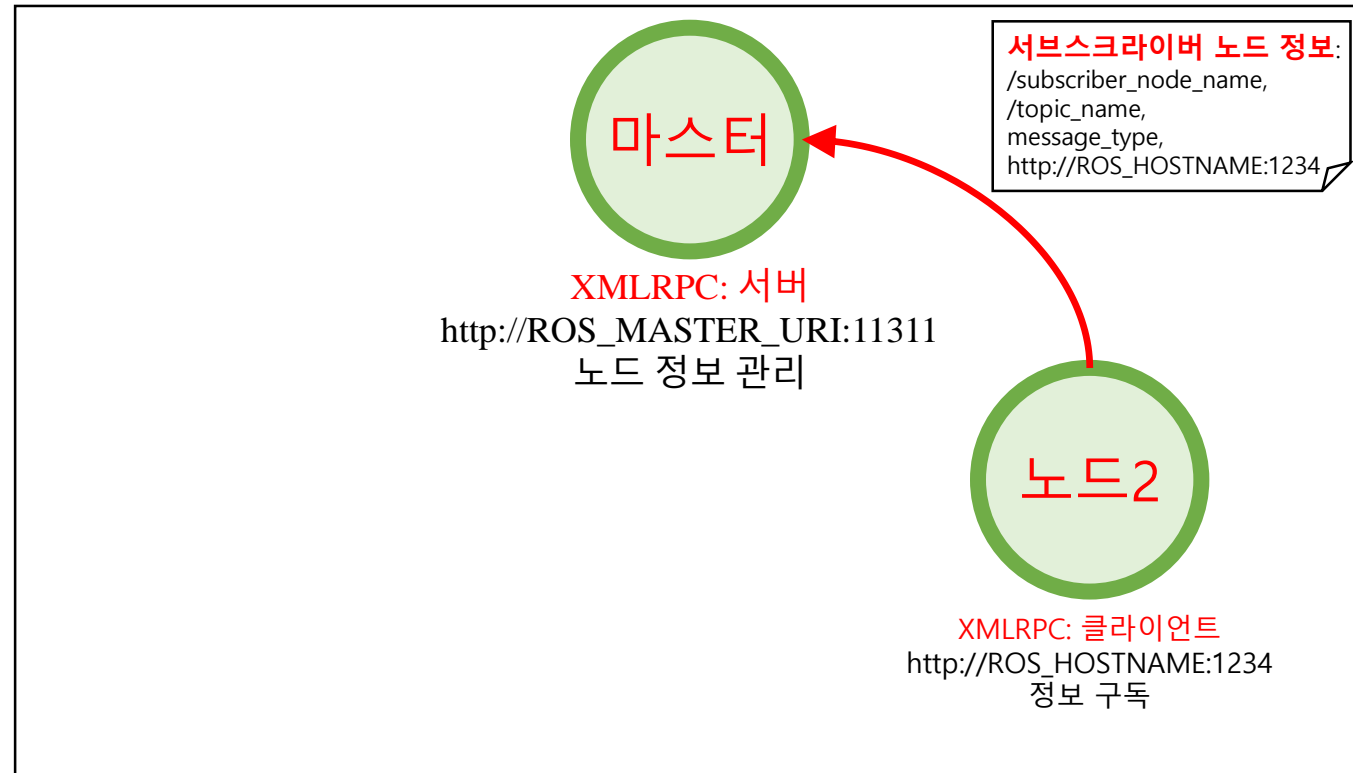
- \$ roscore



# 메시지 통신 개념 잡기!

## 2. 서브스크라이버 노드(Node) 구동

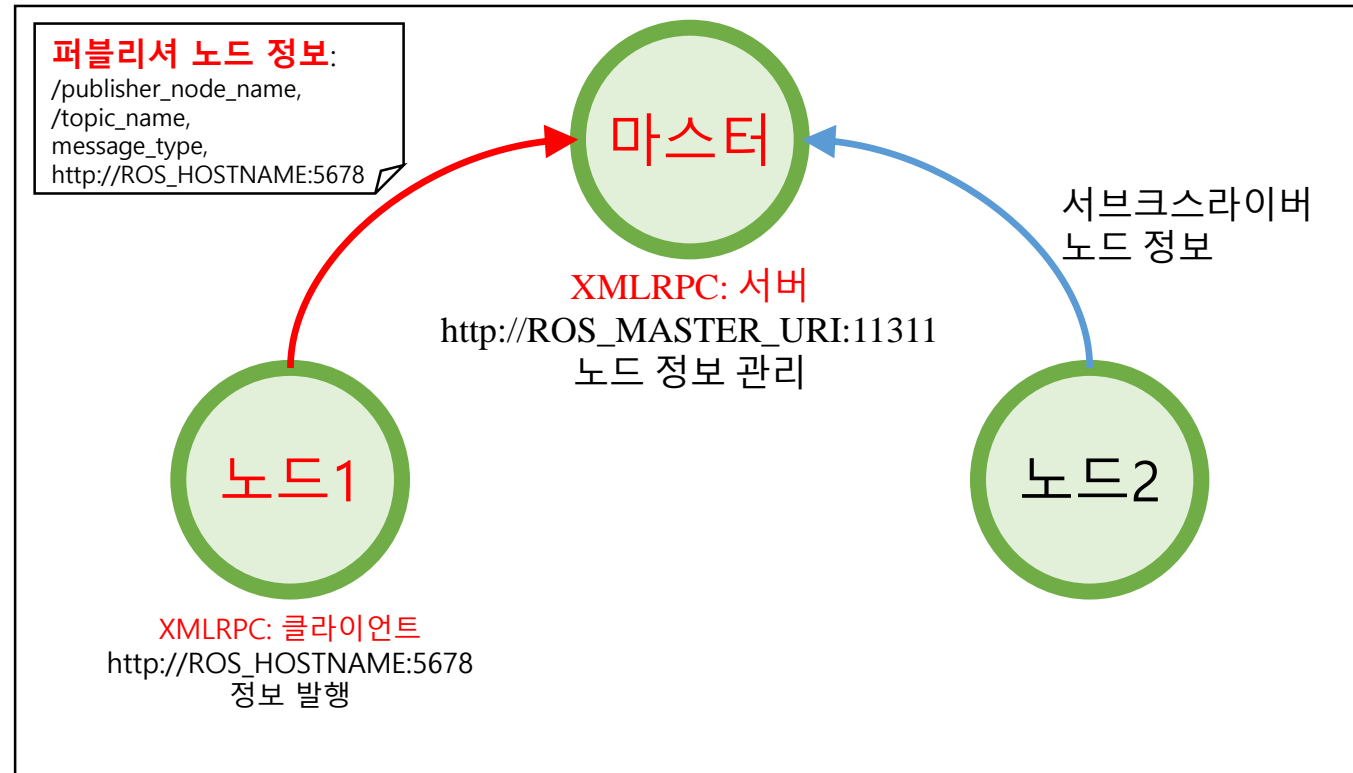
- \$rosvun 패키지이름 노드이름



# 메시지 통신 개념 잡기!

## 3. 퍼블리셔 노드(Node) 구동

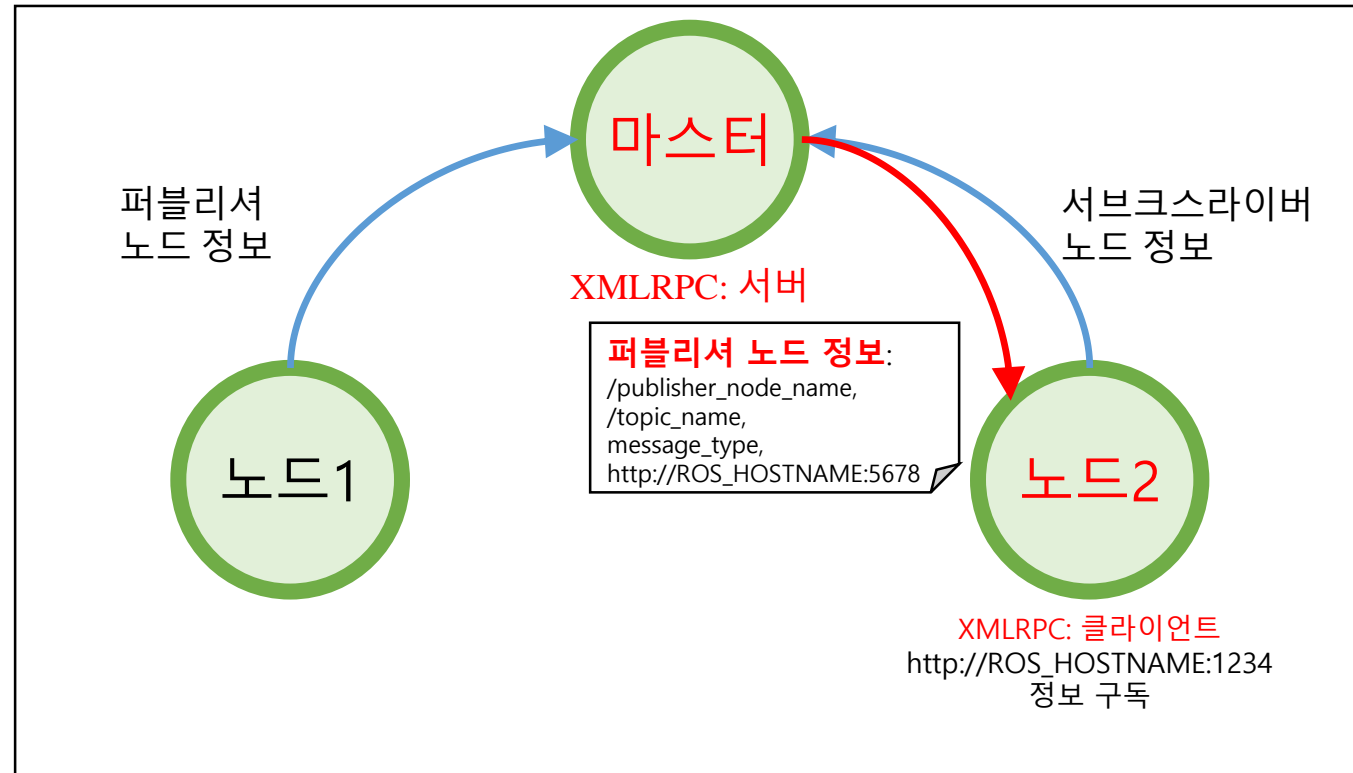
- \$roslaunch 패키지이름 노드이름



# 메시지 통신 개념 잡기!

## 4. 퍼블리셔 정보 알림

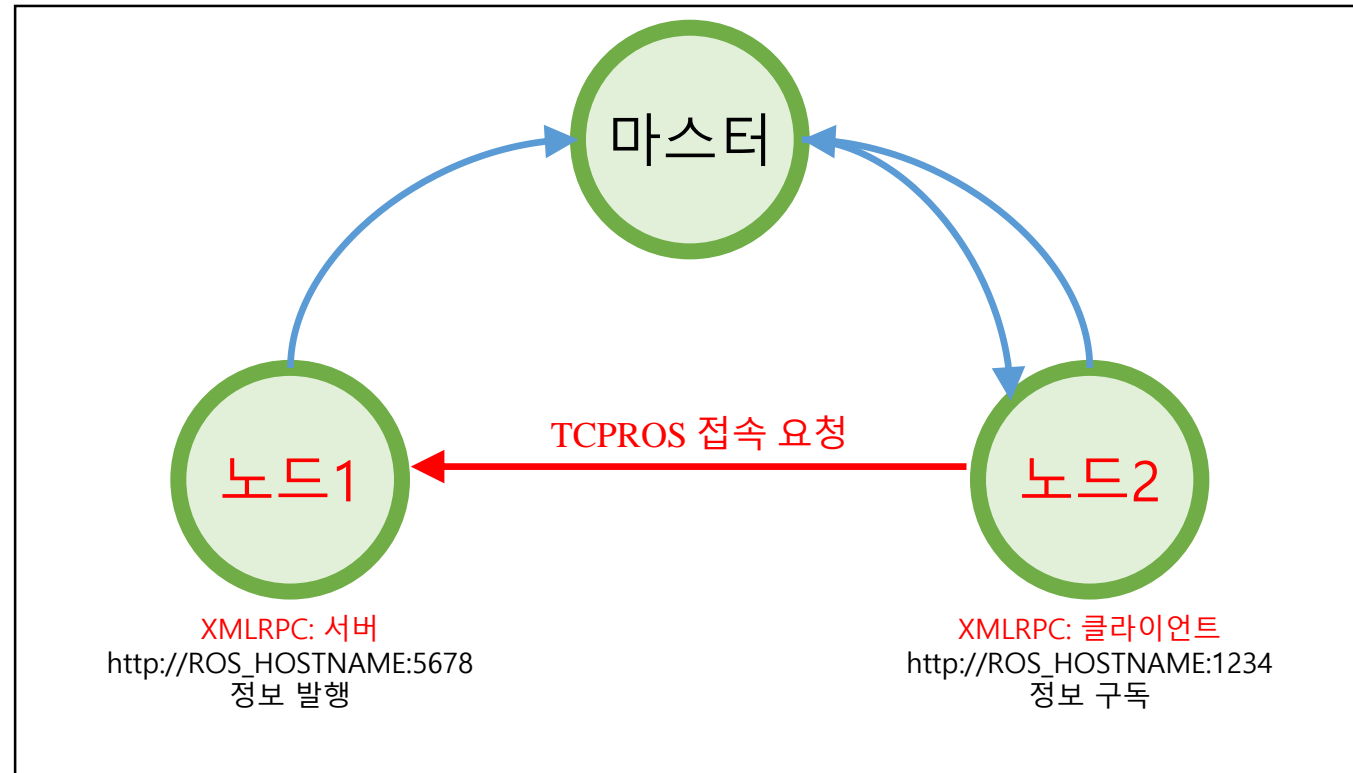
- 마스터는 서브스크라이버 노드에게 새로운 퍼블리셔 정보를 알린다.



# 메시지 통신 개념 잡기!

## 5. 퍼블리셔 노드에 접속 요청

- 마스터로부터 받은 퍼블리셔 정보를 이용하여 TCPROS 접속을 요청

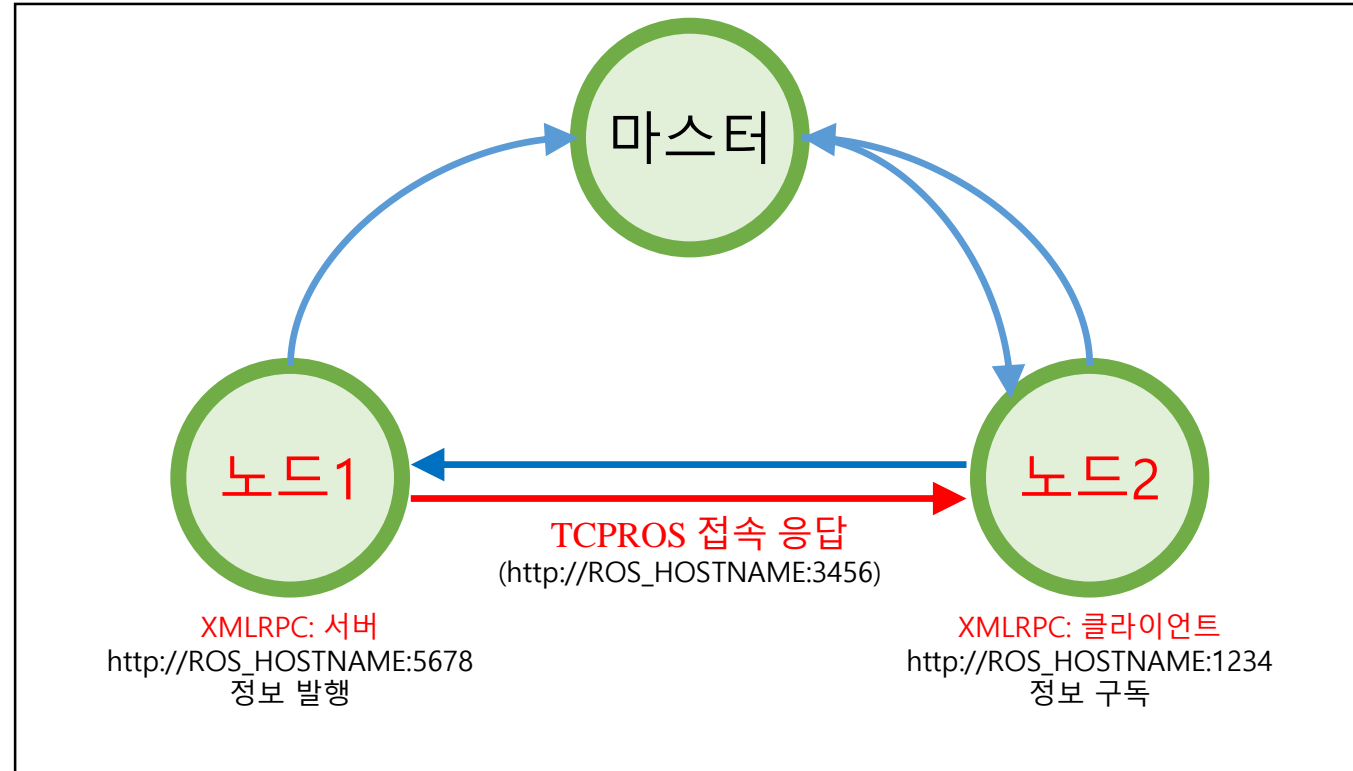




# 메시지 통신 개념 잡기!

## 6. 서브스크라이버 노드에 접속 응답

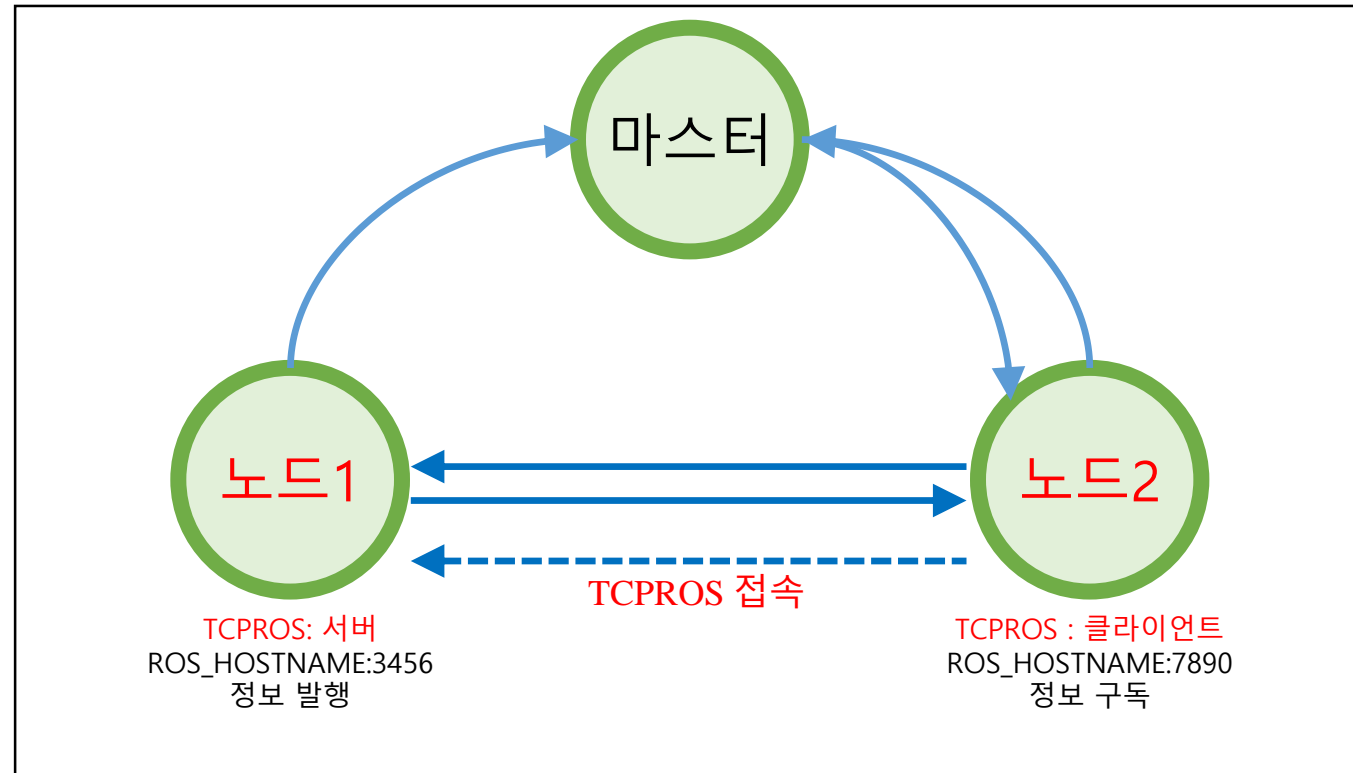
- 접속 응답에 해당되는 자신의 TCP URI 주소와 포트번호를 전송



# 메시지 통신 개념 잡기!

## 7. TCP 접속

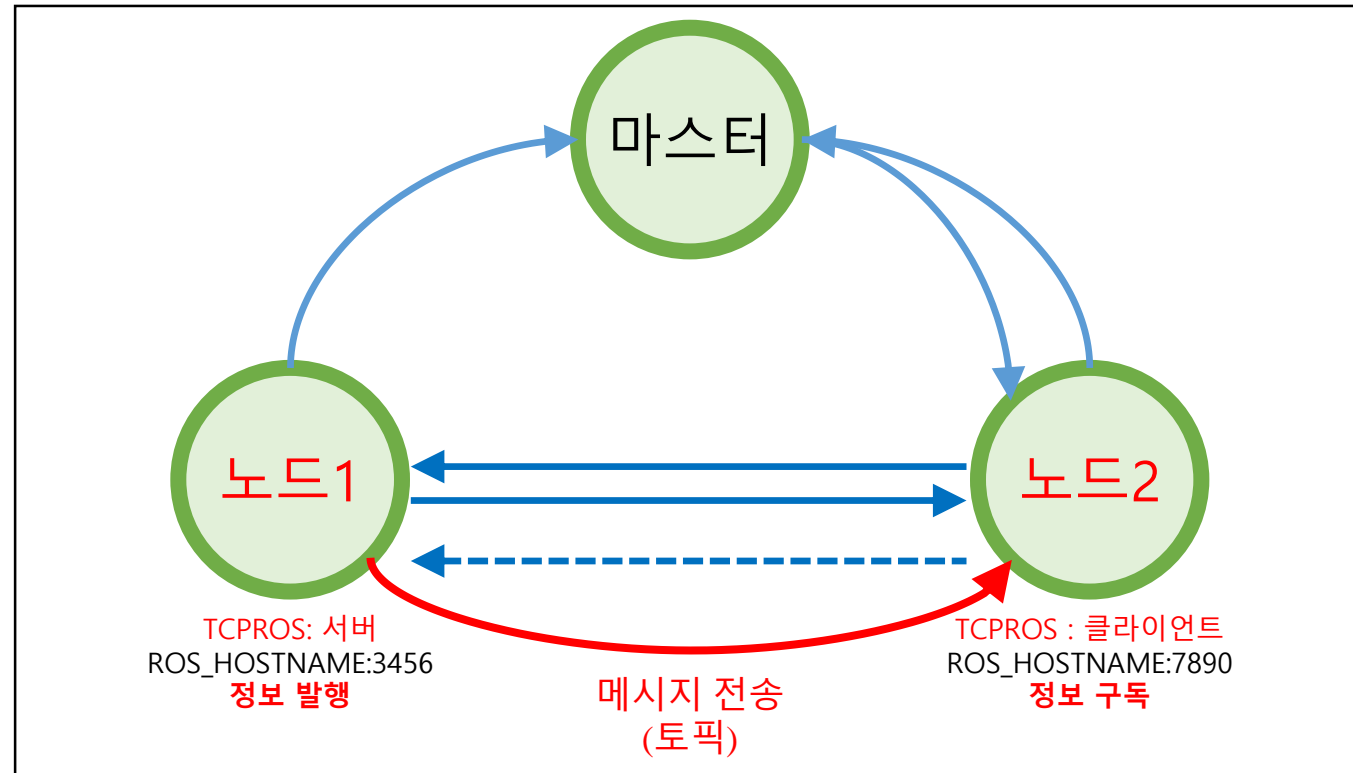
- TCPROS를 이용하여 퍼블리셔 노드와 직접 연결한다.



# 메시지 통신 개념 잡기!

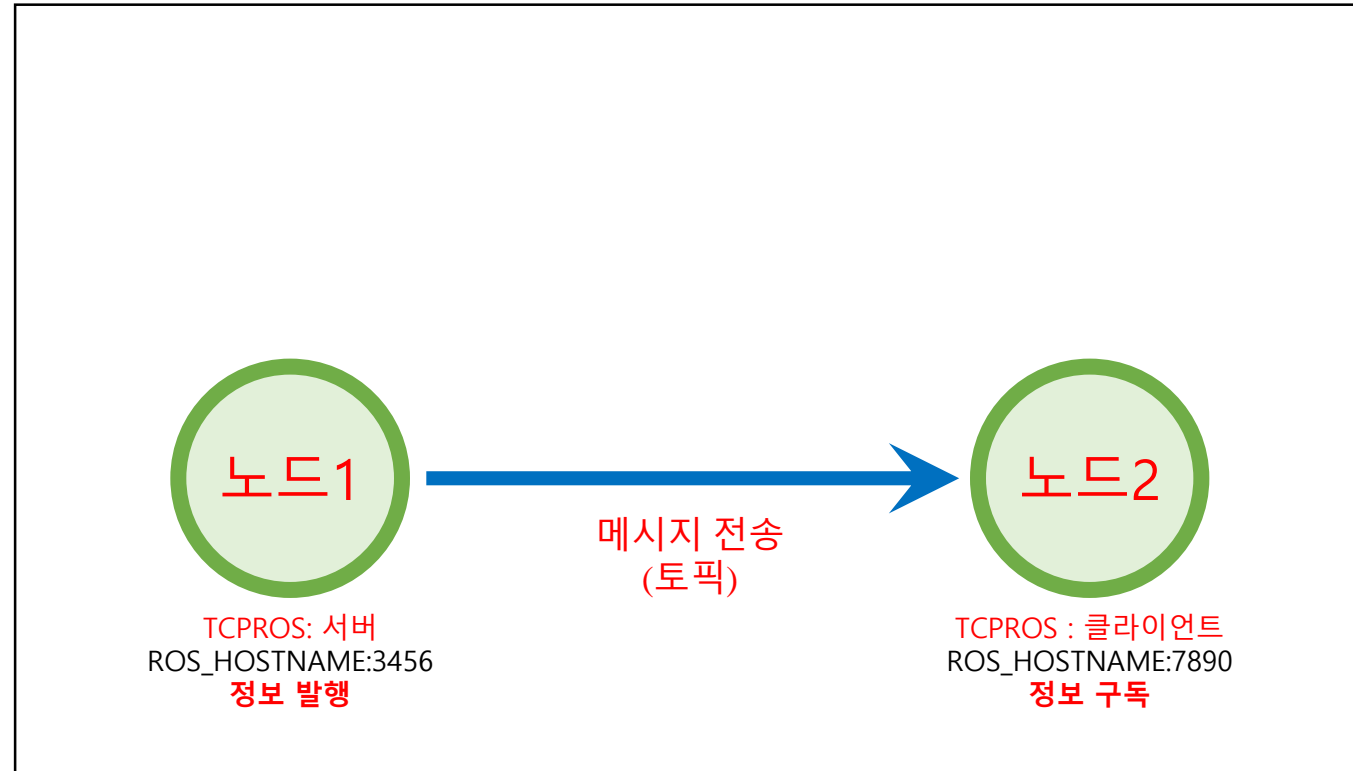
## 8. 메시지 전송

- 발행자 노드는 서브스크라이버 노드에게 메시지를 전송 (토픽)



# 메시지 통신 개념 잡기!

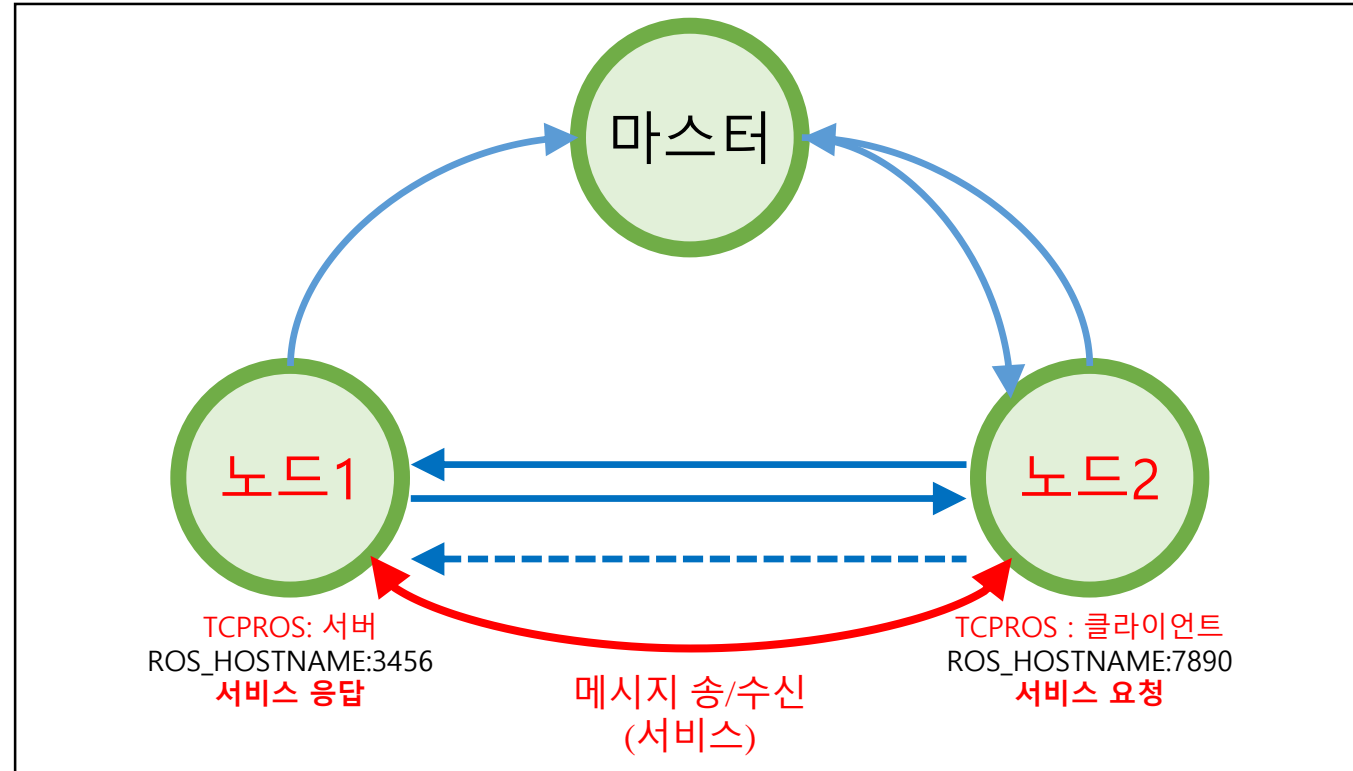
- 토픽방식에서는 접속을 끊지 않는 이상 지속적으로 메시지를 전송한다. 즉, 연속성.



# 메시지 통신 개념 잡기!

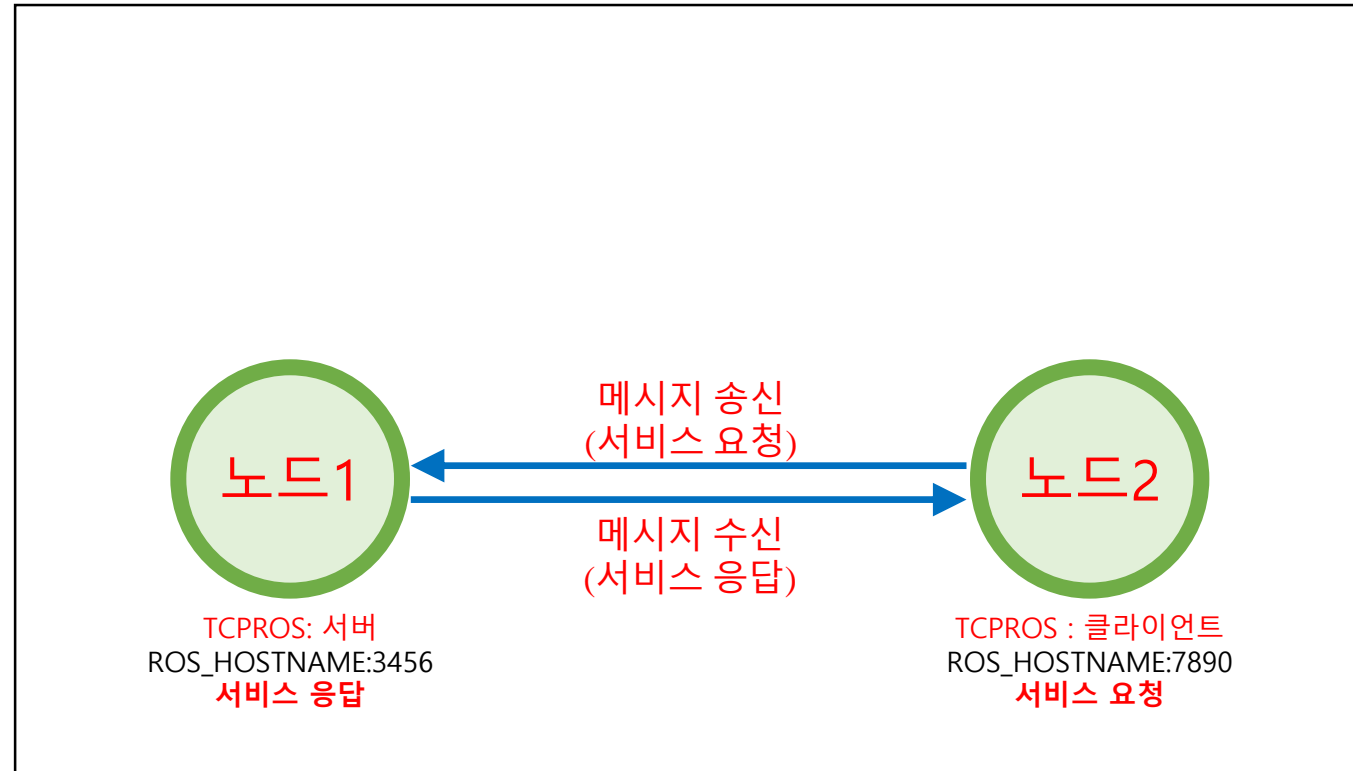
## 9. 서비스 요청 및 응답

- 1회에 한해 접속, 서비스 요청 및 서비스 응답이 수행되고 서로간의 접속을 끊는다.



# 메시지 통신 개념 잡기!

- 서비스는 토픽과 달리 1회에 한해 접속하고 서비스 요청 및 서비스 응답이 수행한 후 서로간의 접속을 끊는다. 즉, 1회성이다.



다시 한번 더 정리!

# 메시지 통신 개념 잡기!

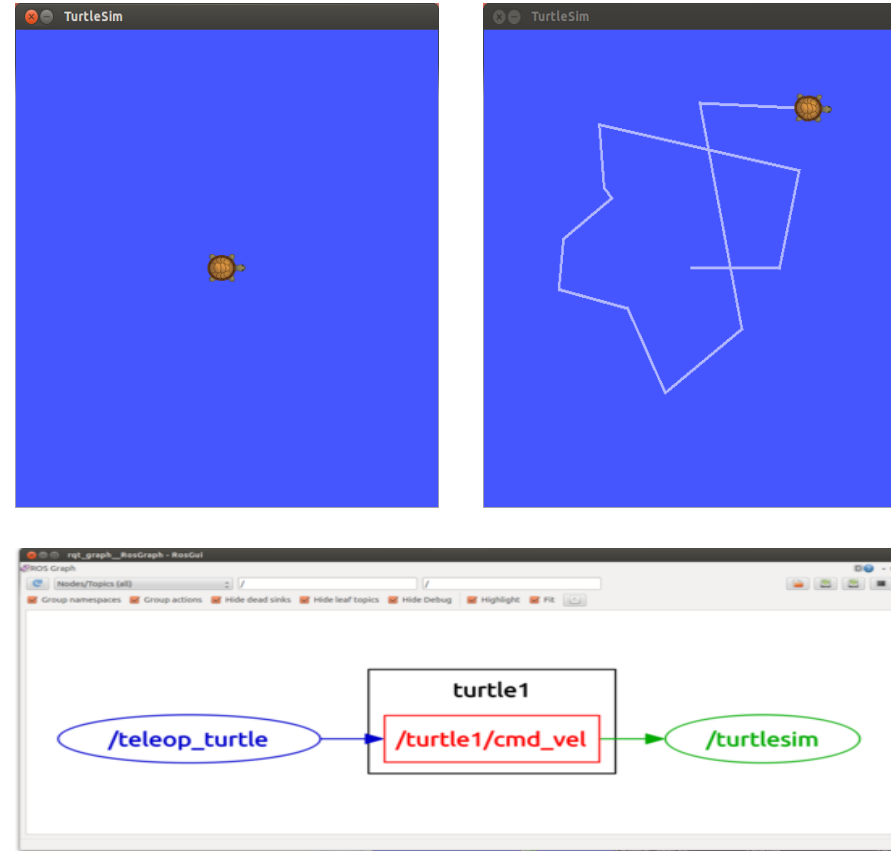
- turtlesim 패키지

- roscore

- rosrun turtlesim turtlesim\_node

- rosrun turtlesim turtle\_teleop\_key

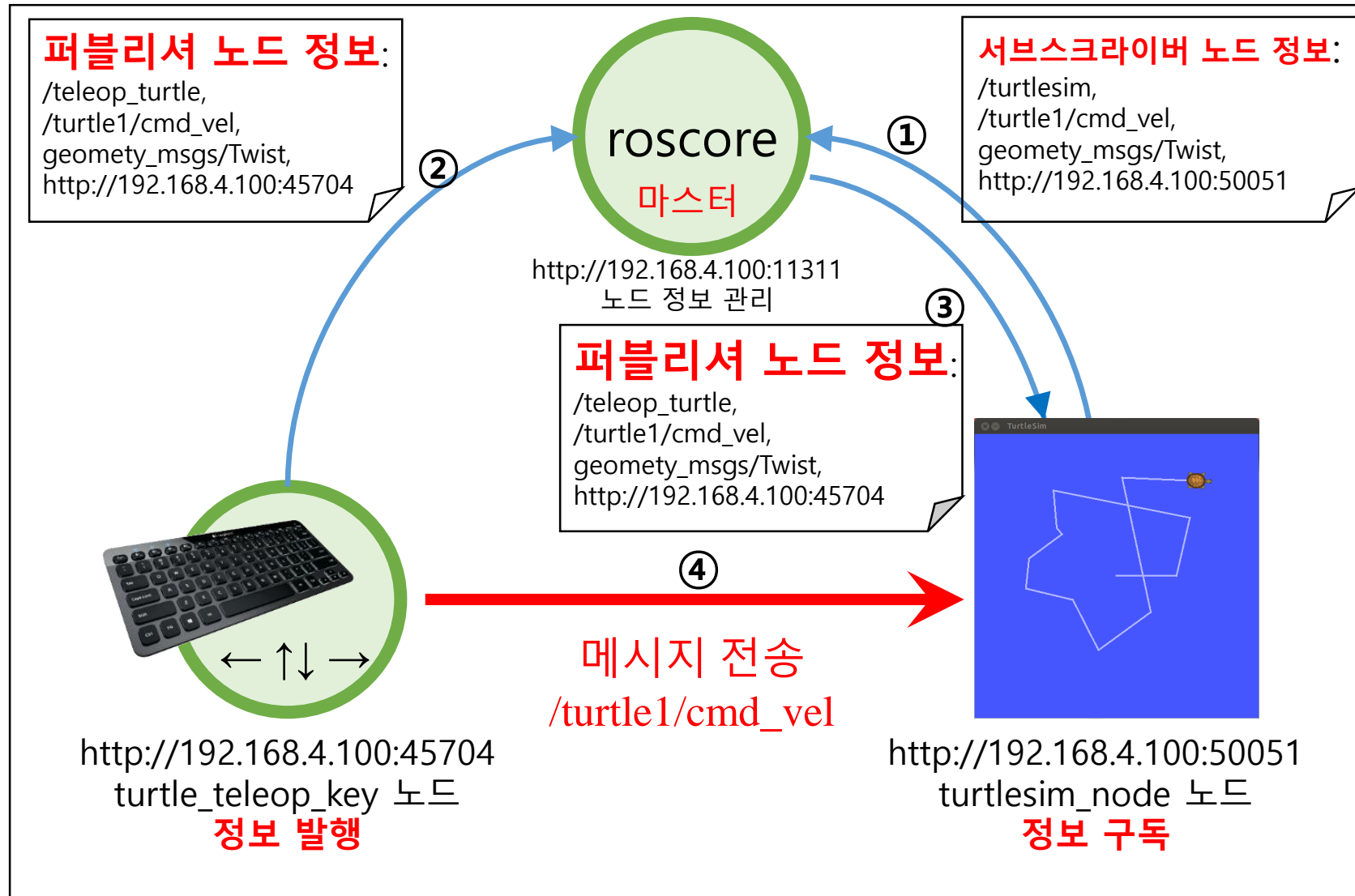
- rosrun rqt\_graph rqt\_graph





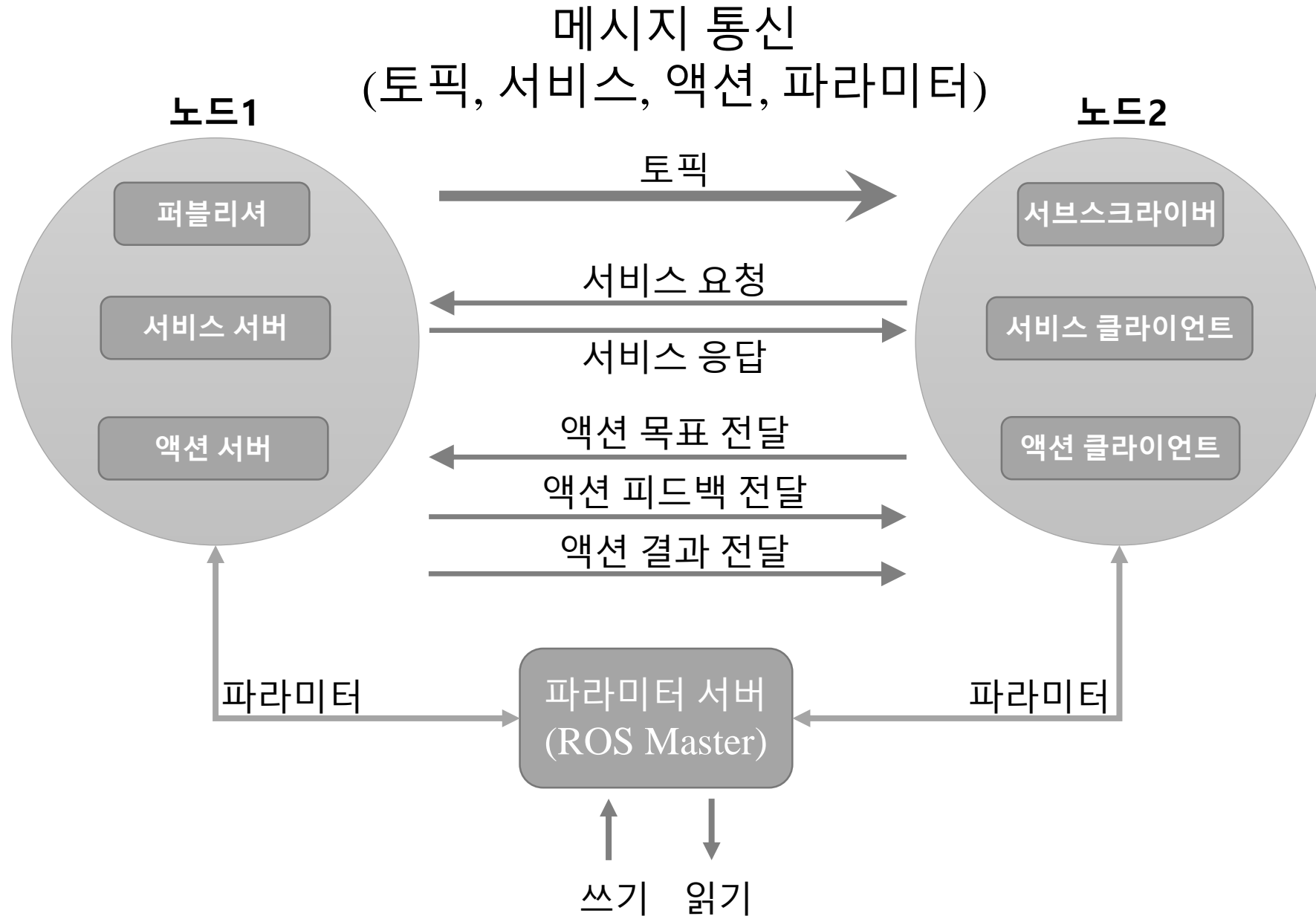
# 메시지 통신 개념 잡기!

## • 10. 예제! turtlesim



메시지?

# ROS 메시지



# ROS 메시지

---

- 메시지는 노드 간에 데이터를 주고받을 때 사용하는 데이터의 형태
  - 토픽, 서비스, 액션은 모두 메시지를 사용
    - <http://wiki.ros.org/msg>
    - [http://wiki.ros.org/common\\_msgs](http://wiki.ros.org/common_msgs)
  - **단순 자료형**
    - 예) 정수(integer), 부동 소수점(floating point), 불(boolean)
    - [http://wiki.ros.org/std\\_msgs](http://wiki.ros.org/std_msgs)
  - **메시지 안에 메시지를 품고 있는 간단한 데이터 구조**
    - 예) geometry\_msgs/PoseStamped
    - [http://docs.ros.org/api/geometry\\_msgs/html/msg/PoseStamped.html](http://docs.ros.org/api/geometry_msgs/html/msg/PoseStamped.html)
  - **메시지들이 나열된 배열과 같은 구조**
    - 예) float32[ ] ranges
    - 예) sensor\_msgs/LaserScan
    - [http://docs.ros.org/api/sensor\\_msgs/html/msg/LaserScan.html](http://docs.ros.org/api/sensor_msgs/html/msg/LaserScan.html)

# ROS 메시지 (예: geometry\_msgs/Twist)



**[geometry\_msgs/Twist]**

**Vector3 linear**  
**Vector3 angular**

**[geometry\_msgs/Vector3]**

**float64 x**  
**float64 y**  
**float64 z**

**[geometry\_msgs/Vector3]**

**float64 x**  
**float64 y**  
**float64 z**

Name, TF

client Library

이기종 디바이스 간의 통신

# 네임(Names)

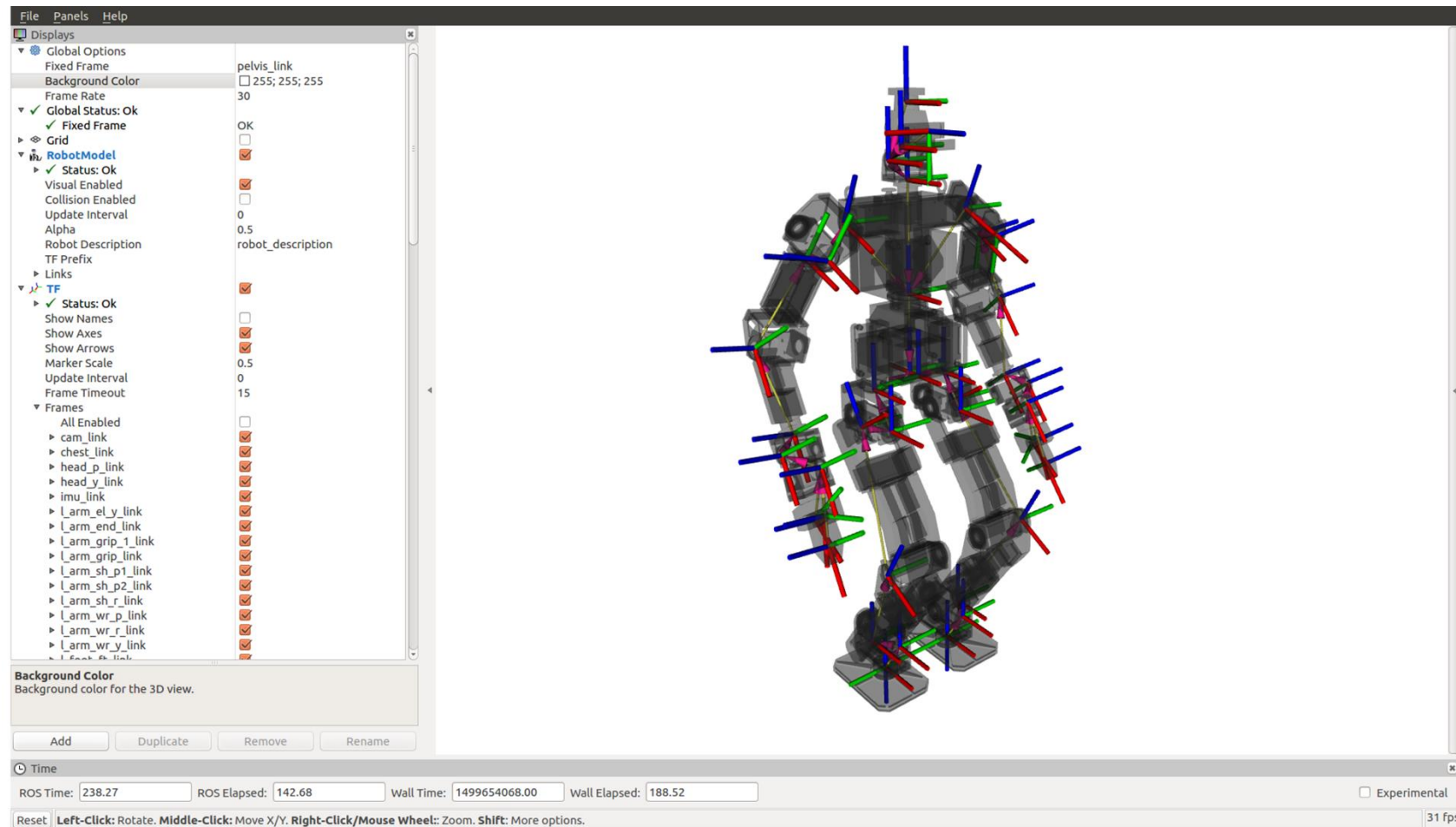
- 네임

- 노드, 메시지(토픽, 서비스, 액션, 파라미터) 가지는 **고유의 식별자**
- ROS는 **그래프**(graph)라는 추상 데이터 형태(abstract data type) 지원
- **글로벌**(global)
  - 문자 없이 네임을 바로 쓰거나 네임 앞에 슬래쉬(/)를 붙임
- **프리베이트**(private)
  - 네임 앞에 틸트(~)를 붙임
- 예제는 7장 ROS 기본 프로그래밍 roslaunch 부분에서 다룸

Node	Relative (default)	Global	Private
/node1	bar -> /bar	/bar -> /bar	~bar -> /node1/bar
/wg/node2	bar -> /wg/bar	/bar -> /bar	~bar -> /wg/node2/bar
/wg/node3	foo/bar -> /wg/foo/bar	/foo/bar -> /foo/bar	~foo/bar -> /wg/node3/foo/bar

# 좌표 변환(TF, transform)

- 각 조인트(joint)들의 상대 좌표 변환
  - 트리(tree) 형태로 조인트들간의 관계도를 표시함
  - 예제는 10장 TF 및 13장 모델링에서 다룸



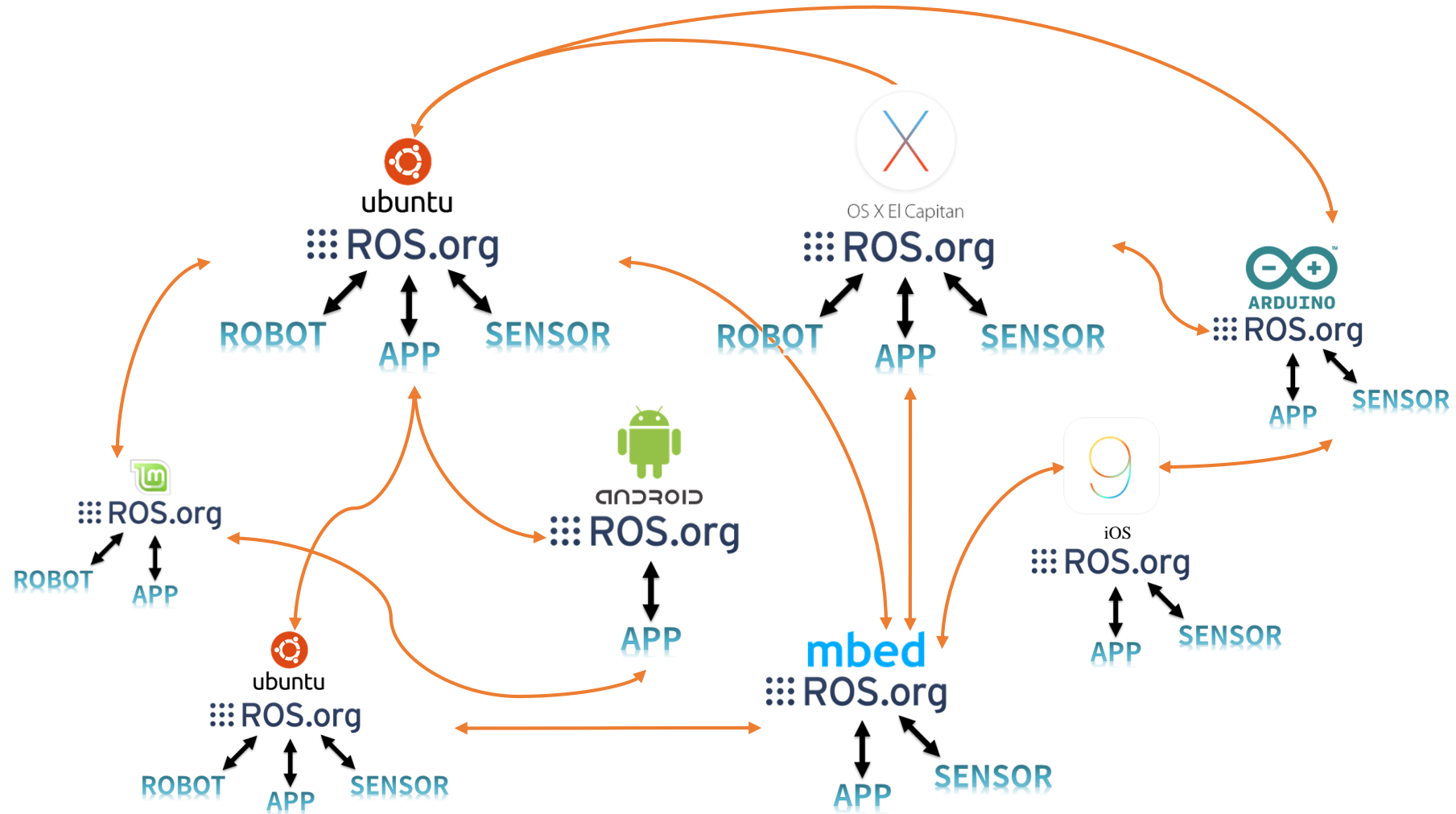


# 클라이언트 라이브러리(Client Library)

- 다양한 프로그래밍 언어 지원
  - roscpp, rospy, roslisp
  - rosjava, roscs, roseus, rosgo, roshask, rosnodejs, RobotOS.jl, roslua, PhaROS, rosR, rosruby, Unreal-Ros-Plugin
  - [MATLAB for ROS](#)
  - [LabVIEW for ROS](#)

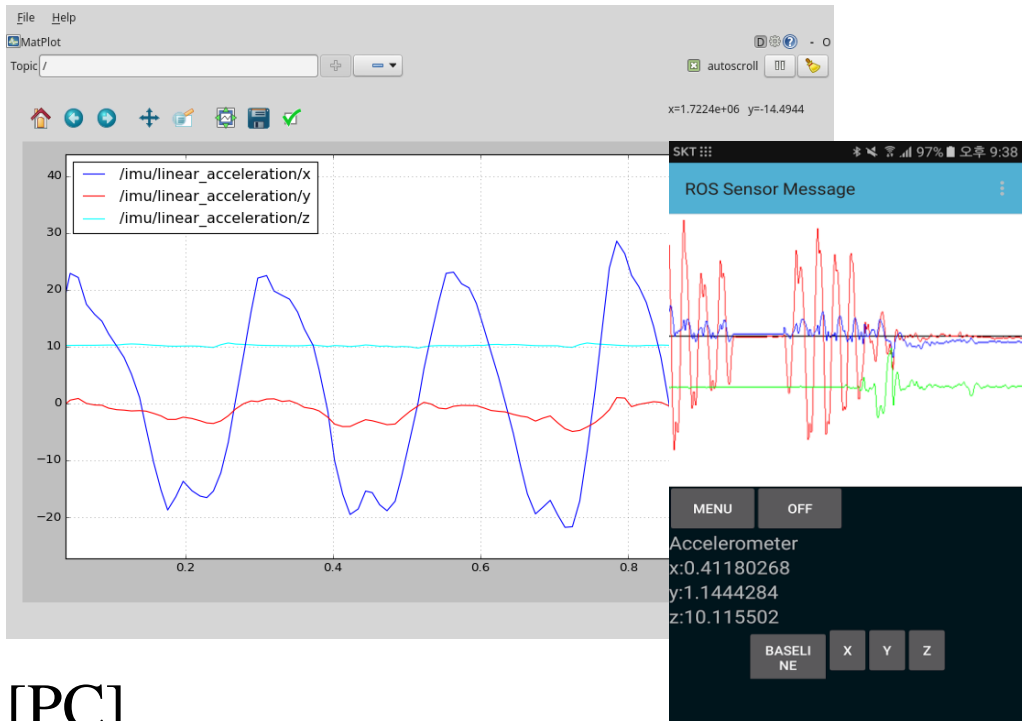


# 이기종 디바이스 간의 통신



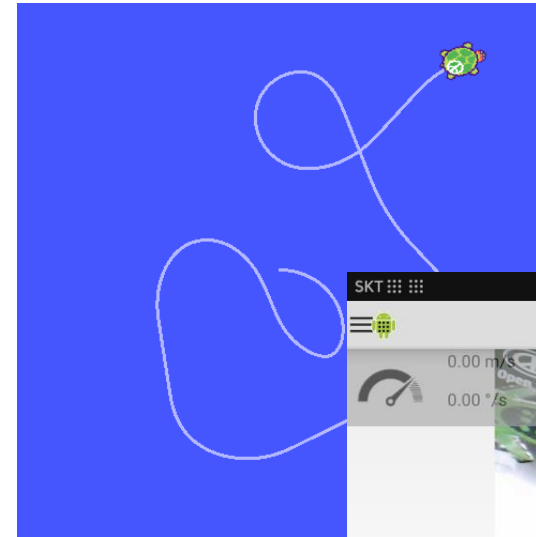
# 이기종 디바이스 간의 통신

- 예제1: 원격으로 이미지 전송 (8장 카메라 참고)
- 예제2: 안드로이드 스마트폰의 가속도 값을 PC에서 확인하기 ([APP](#))
- 예제3: 안드로이드 스마트폰으로 TurtleBot 제어하기 ([APP](#))

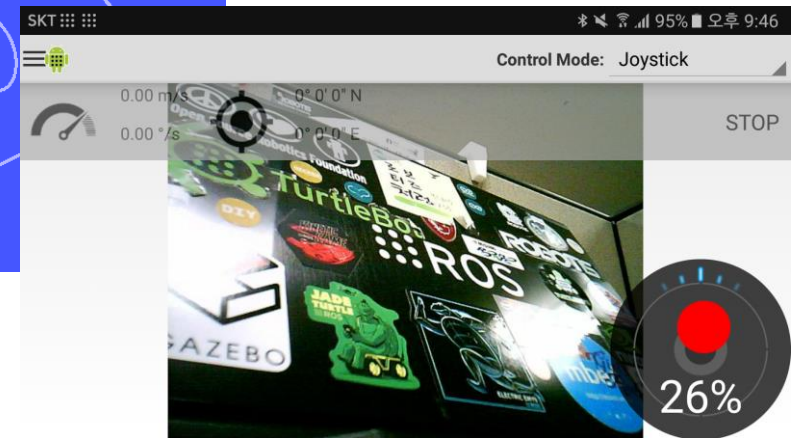


[PC]

[Smartphone]



[PC]



[Smartphone]

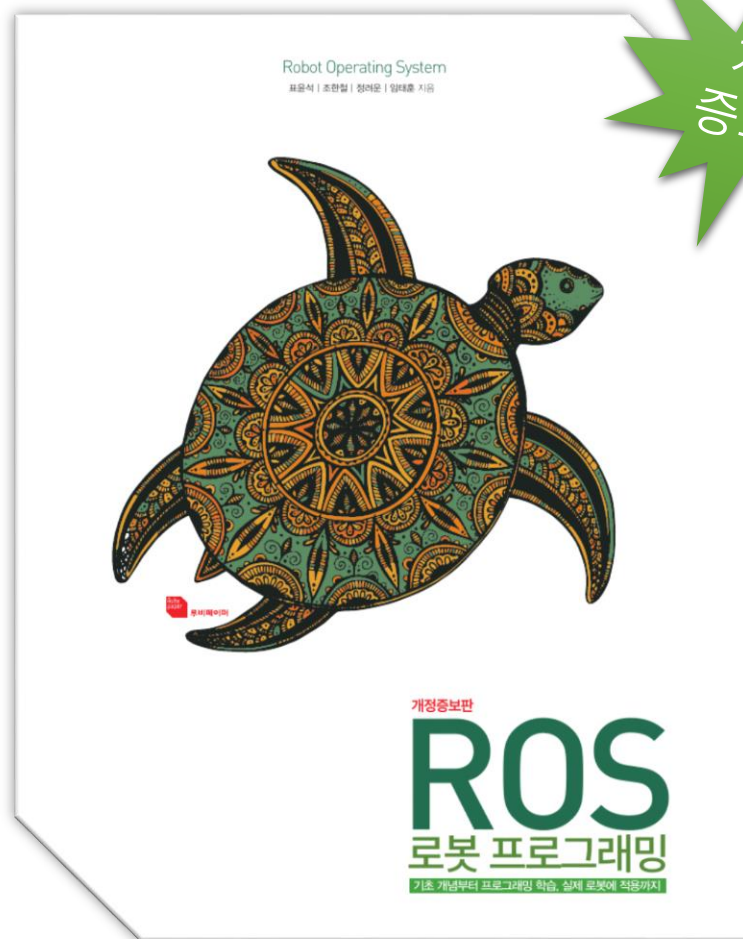
---

# 질문 대환영!

---

\* 온라인 상의 질문이라면  
오로카 및 로열모를 이용해주세요!

여기서! 광고 하나 나가요~



✓ Direct Link

국내 유일! 최초! ROS 참고서!  
ROS 공식 플랫폼 **TurtleBot3** 개발팀이  
직접 저술한 바이블급 ROS 책

# 여기서! 광고 둘 나가요~

## TURTLEBOT3

### 인공지능(AI) 연구의 시작, ROS 교육용 공식 로봇 플랫폼

터틀봇3는 ROS기반의 저가형 모바일 로봇으로  
교육, 연구, 제품개발, 취미 등 다양한 분야에서 활용 할 수 있습니다.

✓ Direct Link



• Collaboration with  

여기서! 광고 셋 나가요~



- ✓ • 오로카
- [www.oroqa.org](http://www.oroqa.org)
- 오픈 로보틱스 지향
- 풀뿌리 로봇공학의 저변 활성화
- 공개 강좌, 세미나, 프로젝트 진행

- ✓ • 로봇공학을 위한 열린 모임 (KOS-ROBOT)
- [www.facebook.com/groups/KoreanRobotics](https://www.facebook.com/groups/KoreanRobotics)
- 로봇공학 통합 커뮤니티 지향
- 일반인과 전문가가 어울러지는 한마당
- 로봇공학 소식 공유
- 연구자 간의 협력

혼자 하기엔 답답하시다고요?

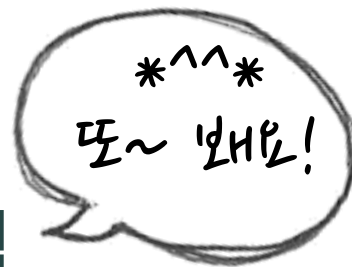
커뮤니티에서 함께 해요~

# 끝.

---

표윤석

Yoonseok Pyo  
pyo@robotis.com  
www.robotpilot.net



[www.facebook.com/yoonseok.pyo](https://www.facebook.com/yoonseok.pyo)