



자율지능연구단 4세부

PX4-ROS2 소개

충북대학교
문성태 교수 (지능로봇공학과)

안녕하세요!

충북대학교 문성태 교수입니다.



안녕하세요!

충북대학교 **문성태** 교수입니다.

2007년~2010년 **국방과학연구소** 연구원:KHP 임무컴퓨터 연구
2010년~2011년 **국가보안기술연구소** 연구원:보안운영체제 연구
2012년~2022년 **한국항공우주연구원** 선임연구원:드론군집 비행기술 연구
2022년~2023년 한국기술교육대학교 조교수
2023년~ 현재 **충북대학교** 조교수



국 방 과 학 연 구 소
Agency for Defense Development

NSR
국가보안기술연구소

KARI
한국항공우주연구원
KOREA AEROSPACE RESEARCH INSTITUTE

자율지능연구단 4세부에서 연구하고 있습니다.

한국항공우주연구원 —
LIG넥스원 —
공통 —

세부 4 과제 목표

무인이동체 통합 자율 지능 협업 시스템 기술 개발



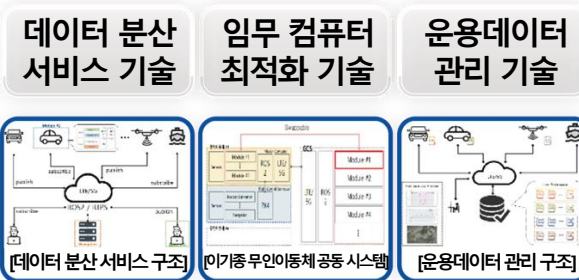
세부 기술 목표 1

시뮬레이터 및 통합 연구 플랫폼 기술



세부 기술 목표 2

자율 지능 협업 시스템



세부 기술 목표 3

인공지능 기반 통합 자율 지능 운용 기술



세부 4 과제 목표

무인이동체 통합 자율 지능 협업 시스템 기술 개발



세부 기술 목표 1
시뮬레이터 및 통합 연구
플랫폼 기술

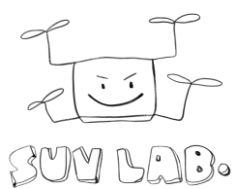


세부 기술 목표 2
자율 지능 협업 시스템



세부 기술 목표 3
인공지능 기반 통합
자율 지능 운용 기술





자율지능연구단 4세부

PX4-ROS2 소개

충북대학교
문성태 (지능로봇공학과)

왜 우리는 ROS2로 가야하는가

01

History



Introduction



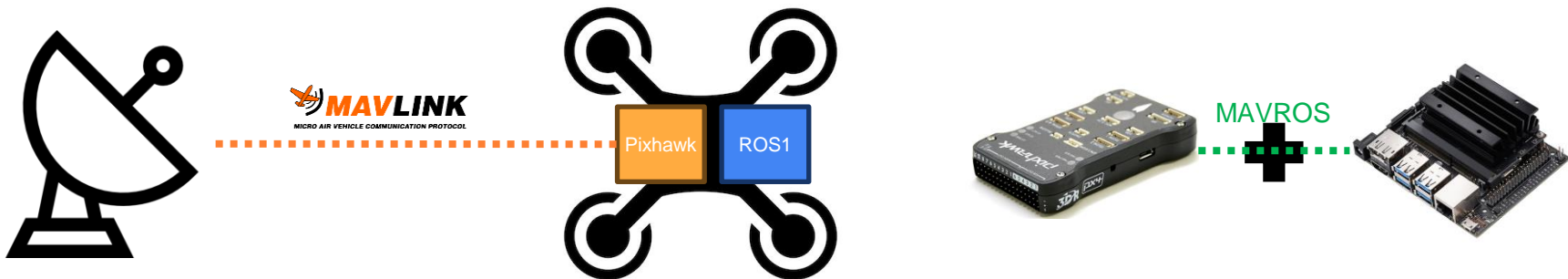
- ❖ 3DR Pixhawk® 1 autopilot
 - Open Hardware 디자인 기반의 Pixhawk 비행제어컴퓨터
- ❖ MAVLink 기반 지상 시스템과의 통신



Introduction



- ❖ 다양한 임무 기능 요구로 임무 컴퓨터 (Mission Computer) 필요
 - Pixhawk + Mission Computer (ex: Jetson nano)
- ❖ 기존에 개발된 임무 알고리즘 활용을 위해 ROS Eco-System 필요
- ❖ ROS와 Pixhawk간의 통신 필요 → MAVROS
 - 하지만, MAVROS의 경우 ROS의 다양한 장점을 활용할 수 없다는 한계가 있음

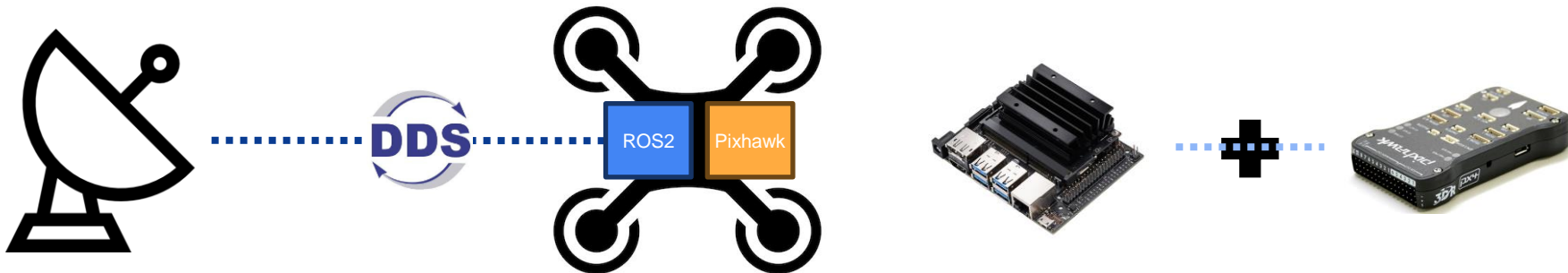


Introduction



❖ 드론 제어 기술의 안정성과 활용 확대로 서비스(임무) 중심 연구가 활발히 진행

- 서비스(임무) 중심 기술 개발을 위해 ROS Eco-System 의존도 증가
- 한편, ROS1 → ROS2로 업그레이드로 인한 통신 강건성 증대
- LTE와 같은 인프라가 구축된 환경에서 MAVLink 보다 많은 기능을 갖는 ROS2의 DDS 통신 기술 관심 증가



Introduction



❖ 하지만, PX4 Society에서는 안정성 문제로 외부 통신은 MAVLink를 사용할

것을 권장
*It's not in scope of this PR, but it needs to be made clear on the documentation how this is supposed to be used. **We do not want people risking their systems by exposing the vehicle internals over-air.** This contribution needs to make clear that its usage needs to be limited to the vehicle scope, and when sharing info over-the-air, **the topics we are exposing are not directly interfacing with the vehicle internals.** – TSC21*

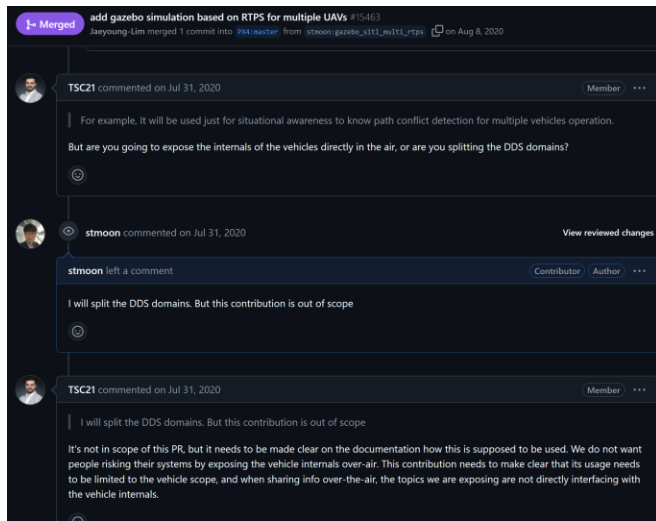


[PX4 System]

VS



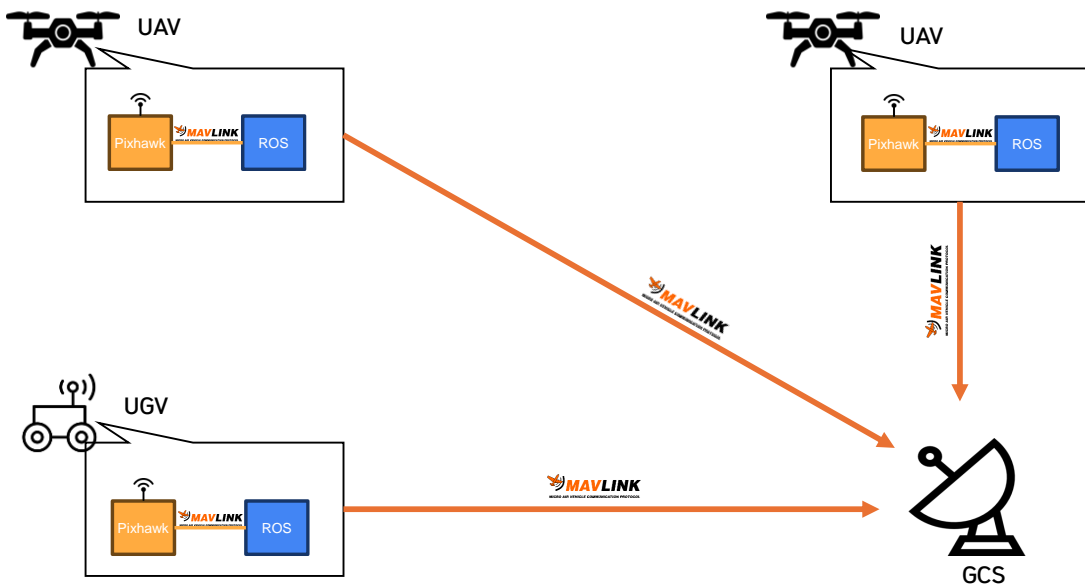
[Proposed System]



MAVLink 중심 통신 시스템



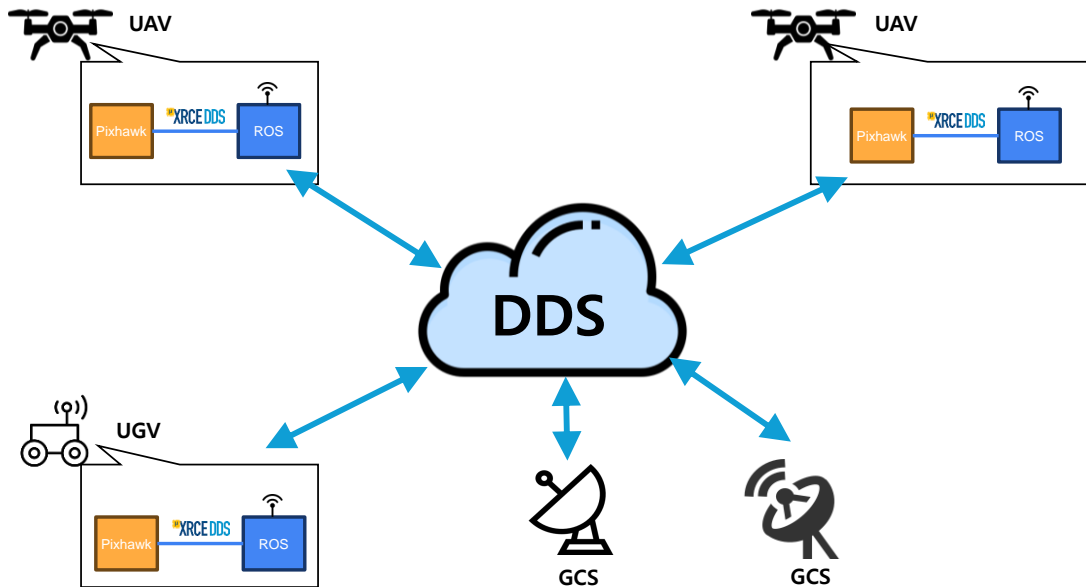
- ❖ MAVLink는 가벼운 프로토콜로 통신 비용이 작지만,
 - 중앙집중형 방식의 문제와 확장성의 한계를 갖는다.



DDS 기반 통신 시스템



- ❖ DDS 기반의 통신은 MAVLink에 비해 다소 통신 비용이 많지만,
 - 분산 네트워크 기반으로 안전하고, 확장성이 좋아 상호 운영성이 높다



Recent PX4



❖ 주요 기능

○ ROS2와의 끊임없는

uXRCE-DDS Interface Enhancements

The uXRCE-DDS interface has received important updates in this release. Notably, the DDS Topics YAML now allows the use of 'subscription_multi' to specify that indicated ROS 2 topics are sent to a separate uORB topic instance reserved for ROS 2. This change enables PX4 to differentiate between updates from ROS and those from PX4 uORB publishers, giving ROS 2 users more control over how their messages interact with existing uORB topics.

Enhanced ROS 2 Integration

PX4 v1.15 introduces an experimental [PX4 ROS 2 Interface Library](#). This C++ library significantly simplifies the process of controlling PX4 from ROS 2 environments. It [provides high-level interfaces for vehicle control and sensor data access](#), abstracting away much of the complexity involved in PX4-ROS 2 communication. This development is a major step towards seamlessly supporting complex robotic system architectures with PX4-powered drones.



PX4 Autopilot Release V1.15: What You Need To Know

by Ramon Roche | Sep 23, 2024 | Announcements, Featured Post | 0 comments



PX4 Autopilot, the open-source flight control system for uncrewed vehicles, [has released its latest stable version](#), v1.15. This update introduces significant improvements for PX4 developers, including tighter integration with ROS 2 and uXRCE-DDS middleware, enhanced hardware support, and navigation improvements. With these enhancements, PX4 v1.15 continues to provide a unified codebase for aerial robotics platforms, including drones, fixed-wing, eVTOLs, and beyond, offering advanced control algorithms and sophisticated sensor fusion capabilities. Let's dive into the key highlights of this release.

Trend (ROSM)



- 미군은 ROS-M (ROS Military) 커뮤니티를 생성하고 ROS 도입을 검토 중

ROS-M
ROS MILITARY PUBLIC WEBSITE

REGISTER LOGIN

ROS-M provides a trusted community of government and industry contributors with the means to collectively cultivate a federated body of re-usable, ROS-compatible software for developing defense robotic systems.

ROS-M is built on the ROS framework

ROS-M provides management tools, such as a searchable master registry for finding existing software packages, as well as custom software, like a ROS to Interoperability Profiles (IOP) gateway, interoperability tools, and processes specific to defense robotics applications.

Rapidly produce prototype systems in less time, for less cost, and with less risk.

ROS-M enables system developers and integrators to readily identify, access, and use existing software for robot applications, as well as readily develop new software packages as needed or desired.

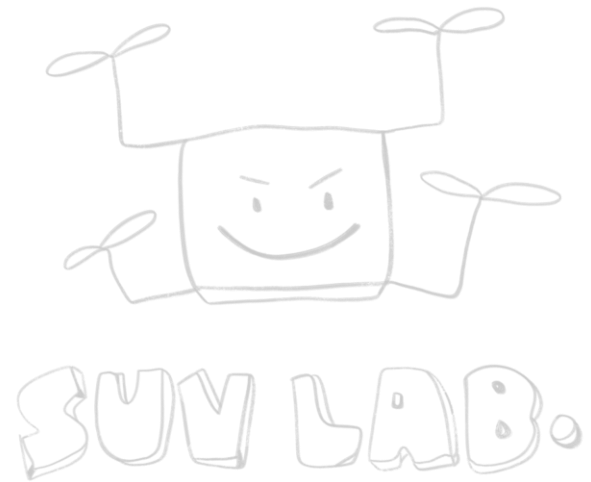
Benefits of ROS-M

- Greater Collaboration**
ROS-M facilitates greater collaboration by relying on a common, de facto standard in ROS that partnering organizations can use as a basis for cooperative software development efforts.
- Increases Innovation**
ROS-M increases innovation by enabling organizations to find their resources at developing new innovative capabilities that build on top of the baseline software functions in the core, common libraries.
- Improves Software Reliability**
ROS-M improves software reliability and security by finding and fixing defects and making improvements to the same, reusable code.

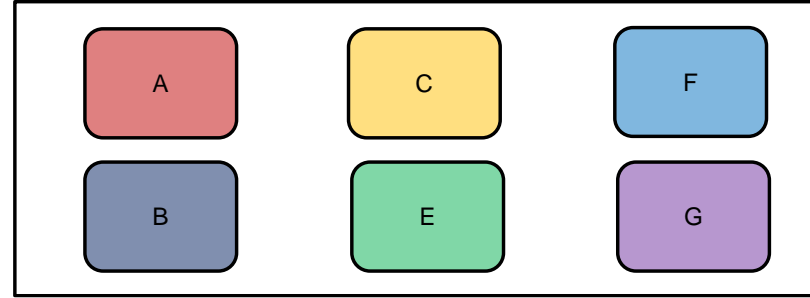


02

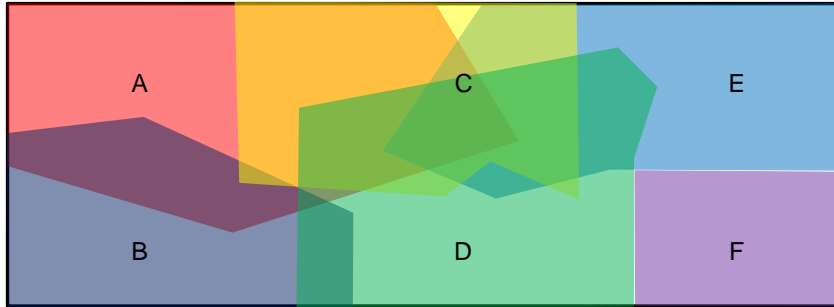
PX4-ROS2



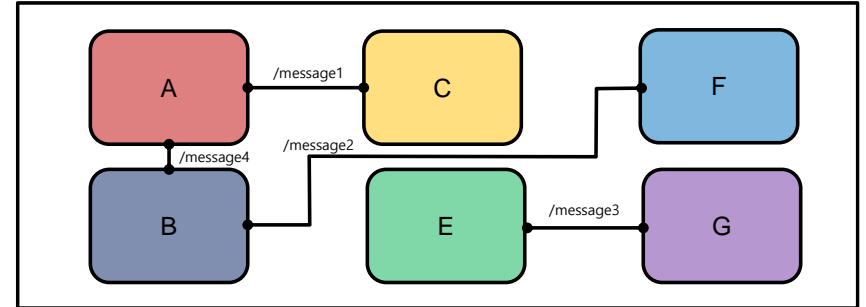
Concept



[Design]



[Output using Legacy S/W Development]



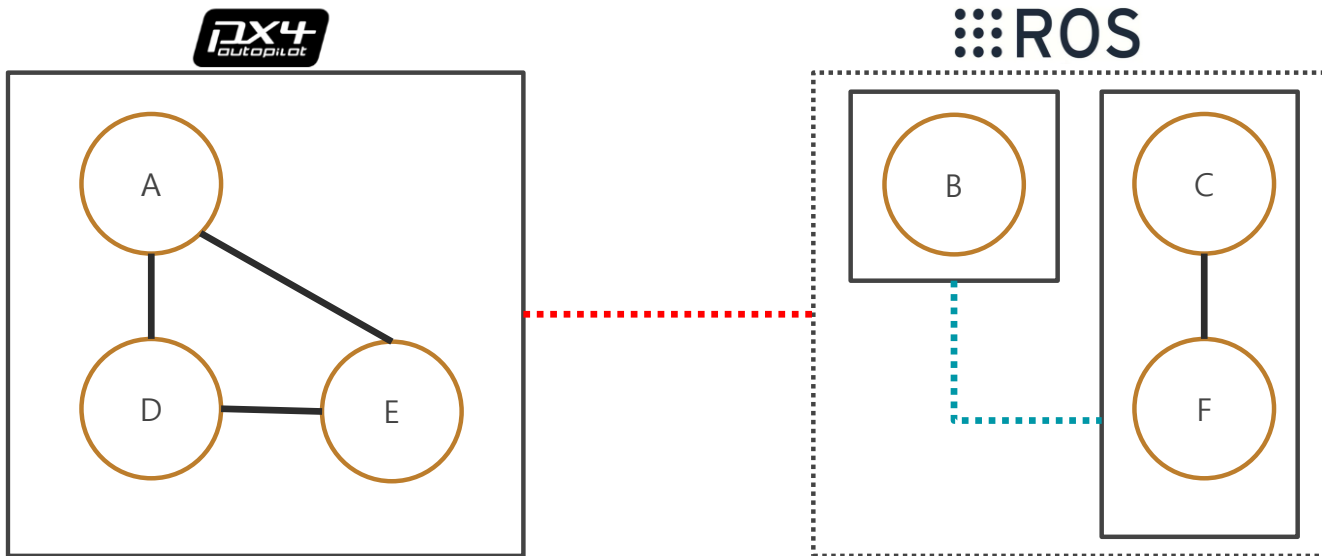
[Output using Component based S/W Development]

PX4 vs ROS

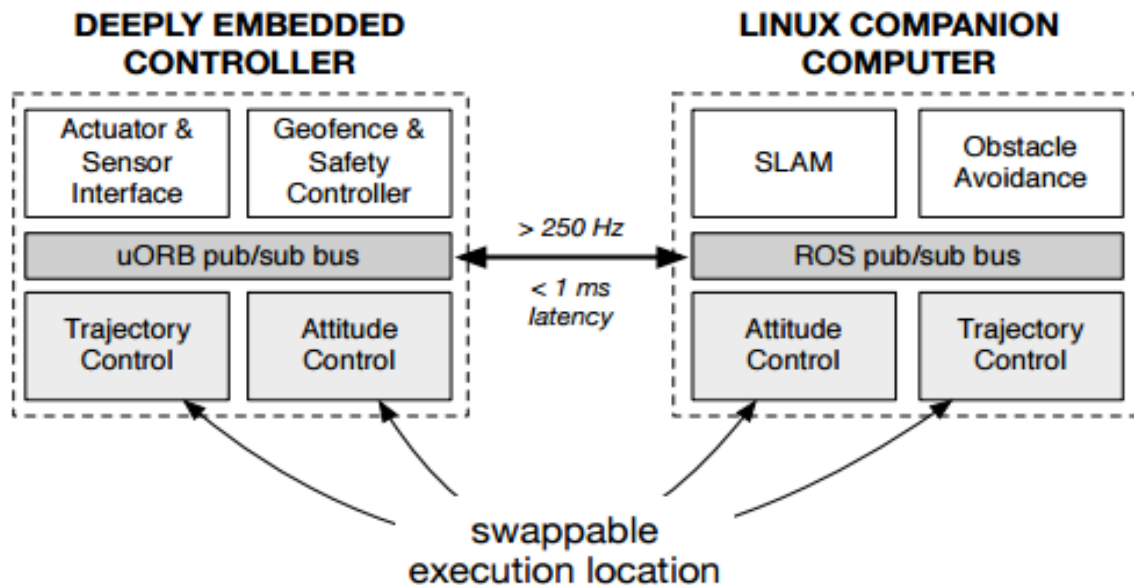



❖ 공통점

- Message Driven Approach
- Component Based Development (CBD)
- Publish-Subscribe Design Pattern



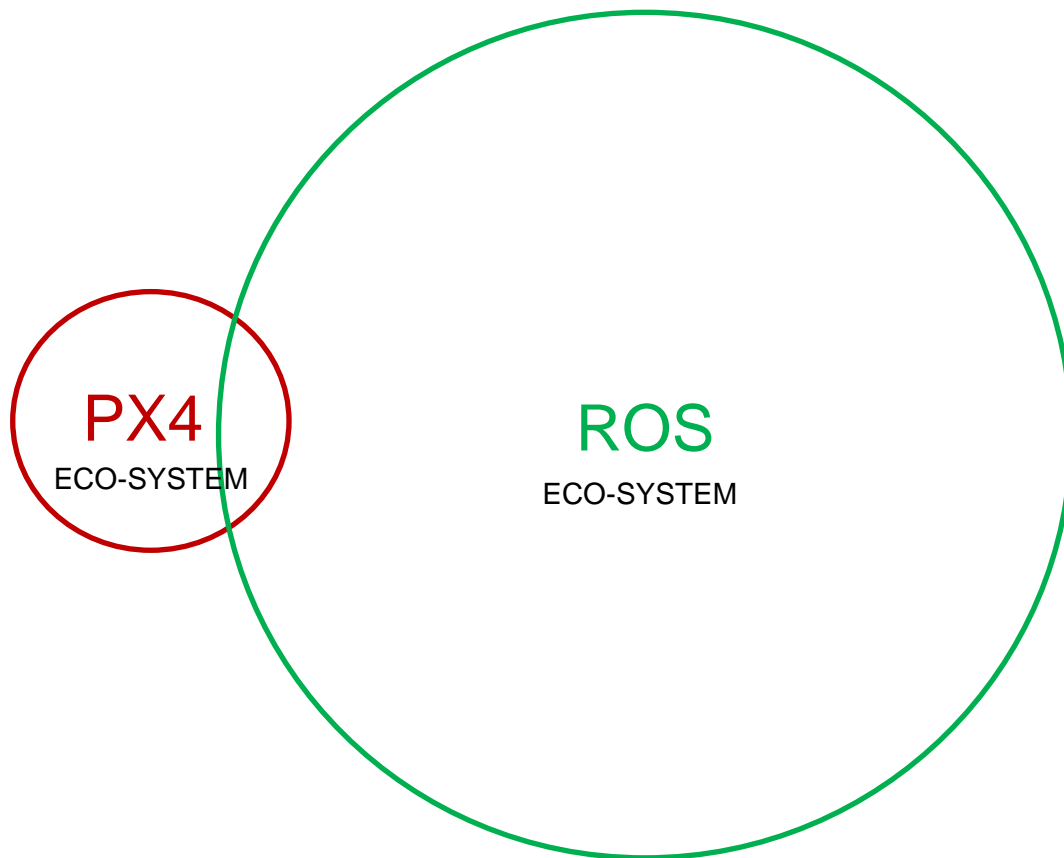
Target



DON'T REINVENT
THE WHEEL 
JUST REDESIGN IT



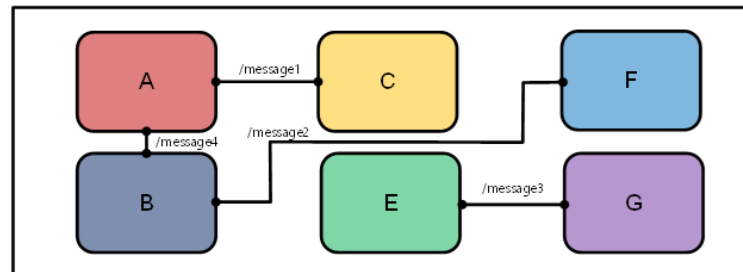
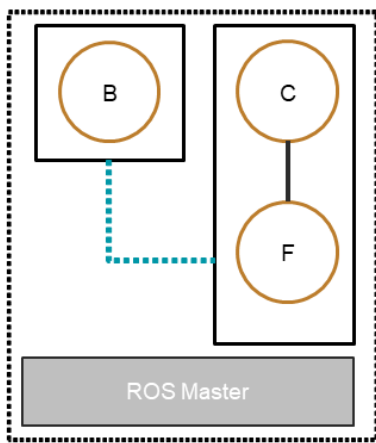
Target



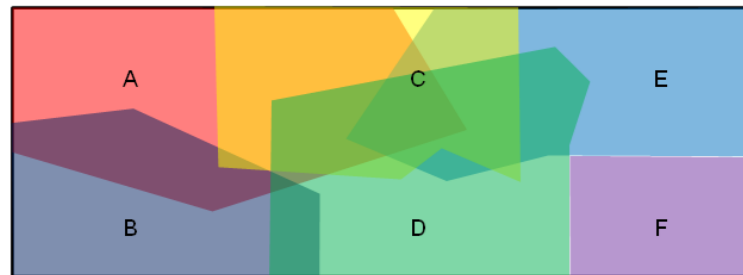
ROS1



- Fault isolation
- Faster development
- Modularity
- Code reusability



VS

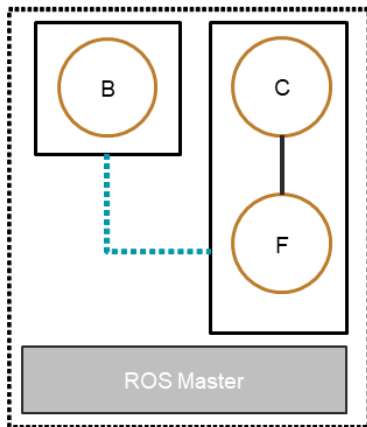


Limitation of ROS1



Possible

- Fault isolation
- Faster development
- Modularity
- Code reusability



Impossible

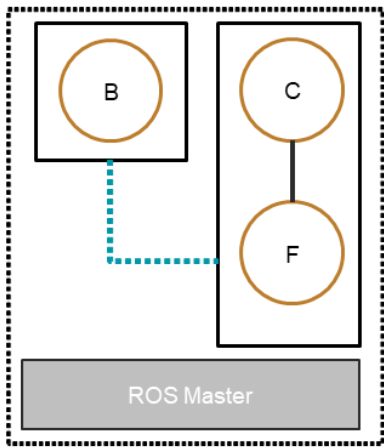
- Real-time
- Embedded system
- Fault tolerance
- Multi-robot (상호 운용성)

ROS1 vs ROS2



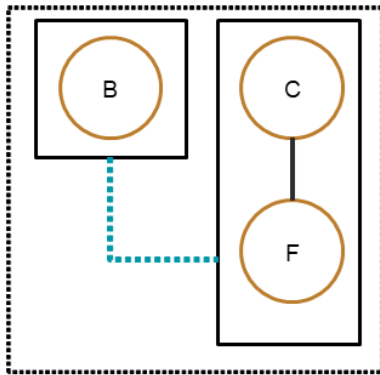
ROS1

- Fault isolation
- Faster development
- Modularity
- Code reusability

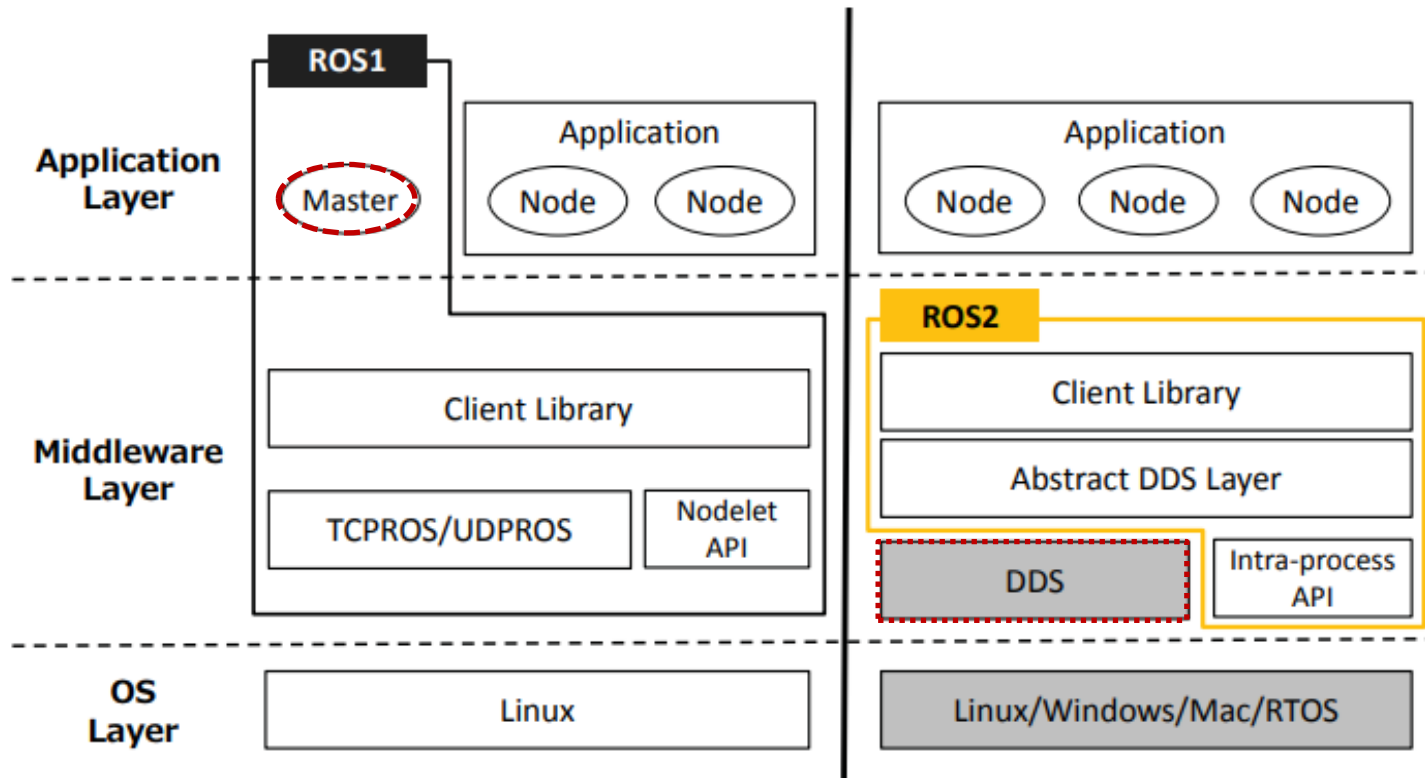


ROS2

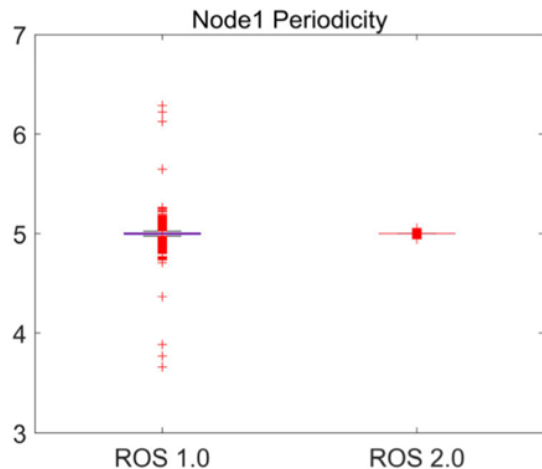
- Fault isolation
- Faster development
- Modularity
- Code reusability
- Real-time
- Embedded system
- Fault tolerance
- Multi-robots (상호 운용성)



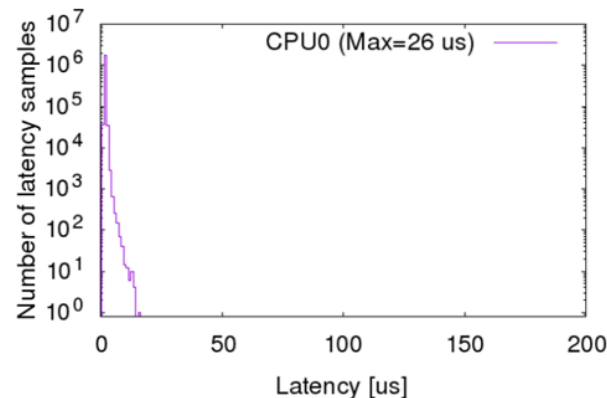
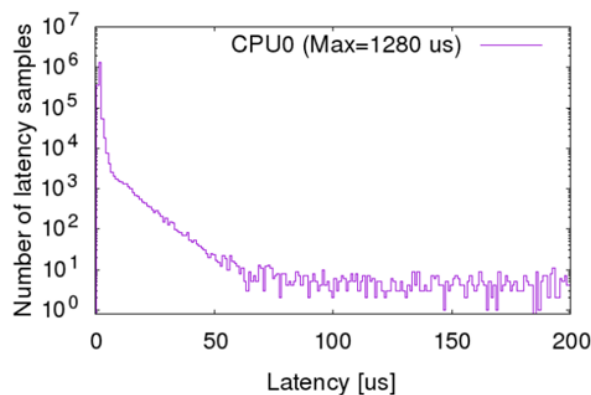
ROS1 → ROS2



Performance



[Periodicity of node]

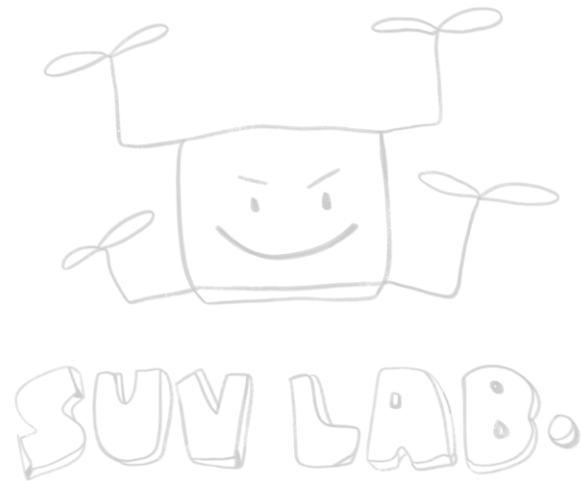


[Scheduling latency in a stressed environment]

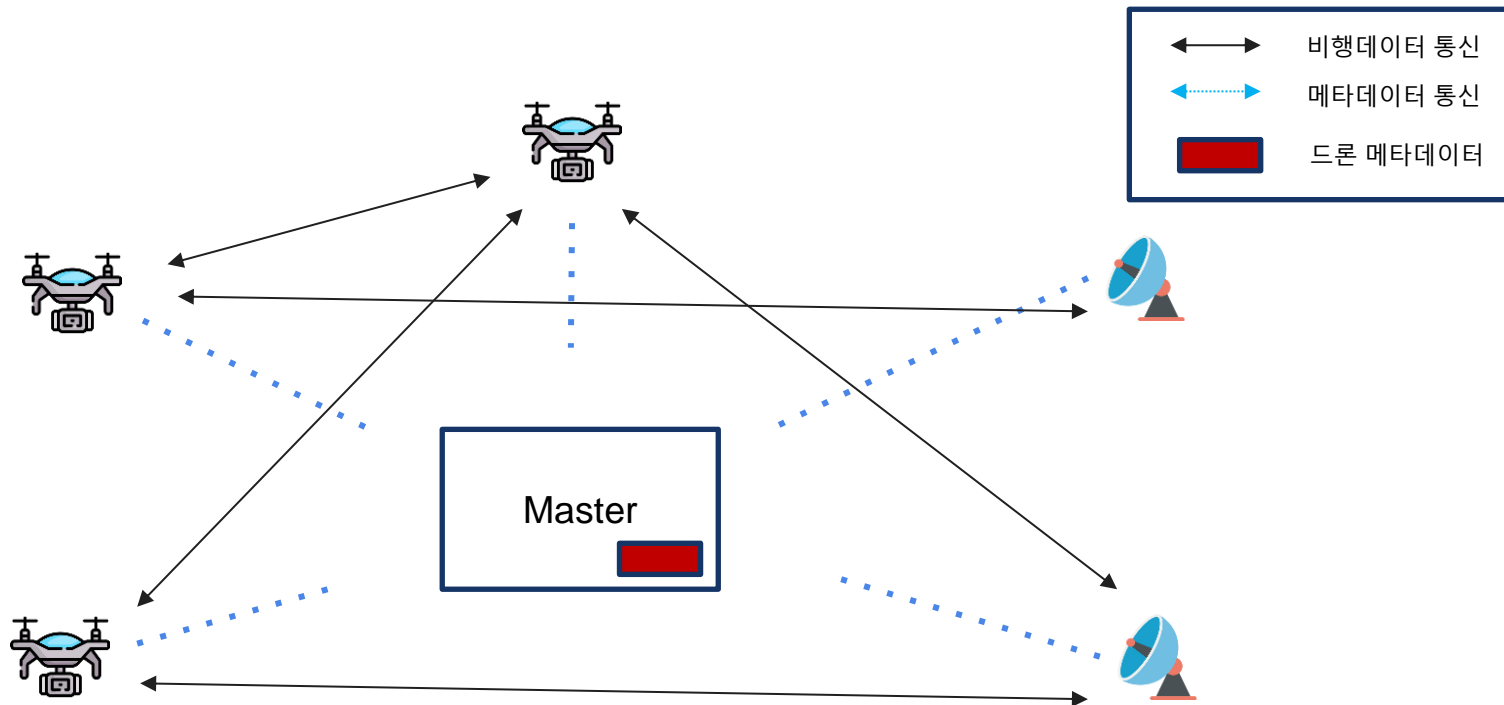
ROS1은 더 이상 지원이 되지 않고,
다수 무인 이동체들의 상호 운영을 위해서는
ROS2를 사용해야 한다.

03

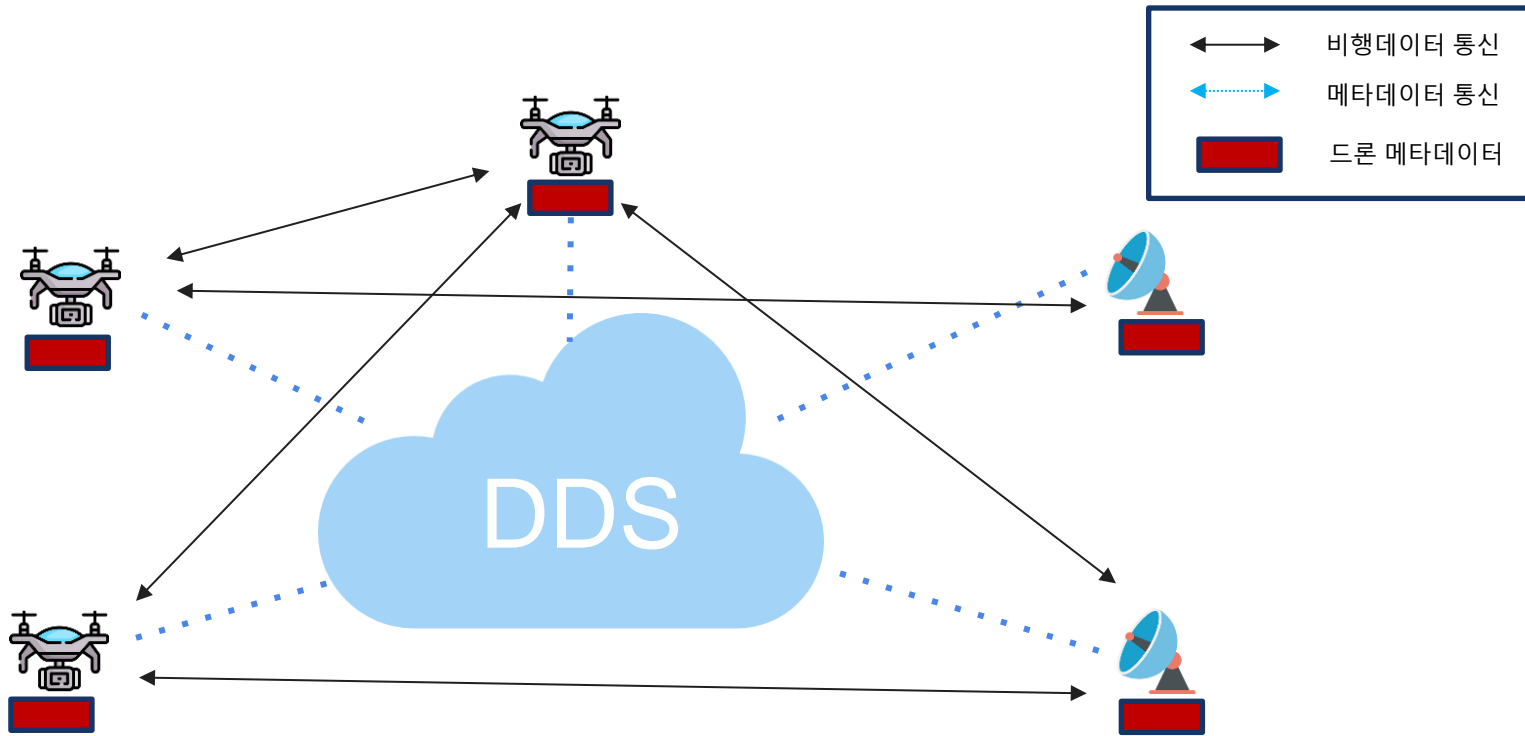
Problem



ROS1

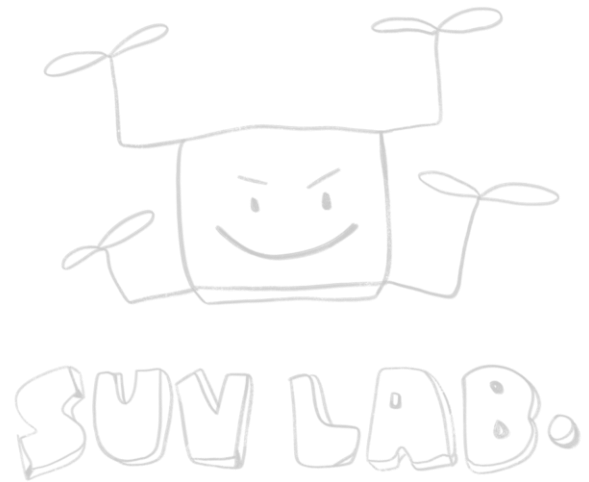


ROS2



04

Solution

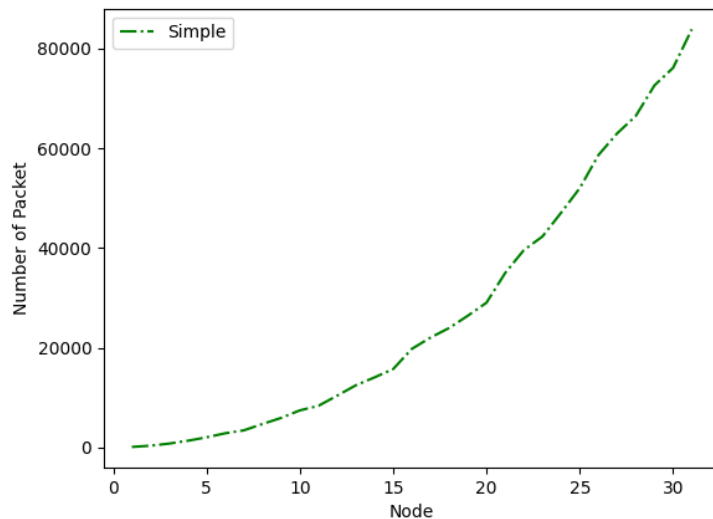
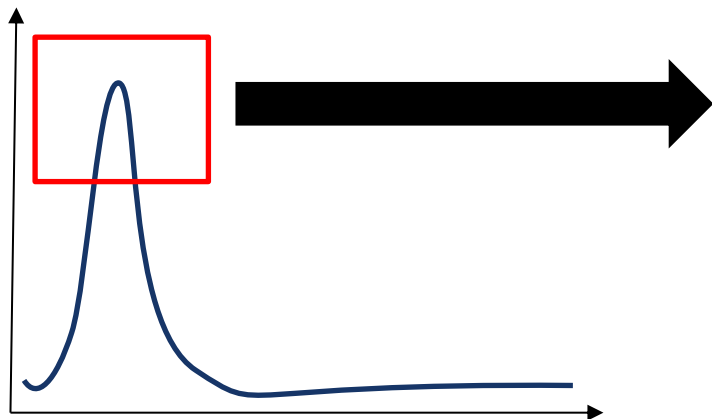


Traffic Analysis for Discovery



❖ 노드에 증가에 따른 DDS 통신 패킷량 분석

- 노드 수가 증가할수록 급격하게 데이터 송수신량 증가 확인
- 무선 네트워크에서 다수 무인 이동체 운영 시 통신 문제 발생 가능



❖ DDS 통신에서 사용되는 패킷 종류 분석

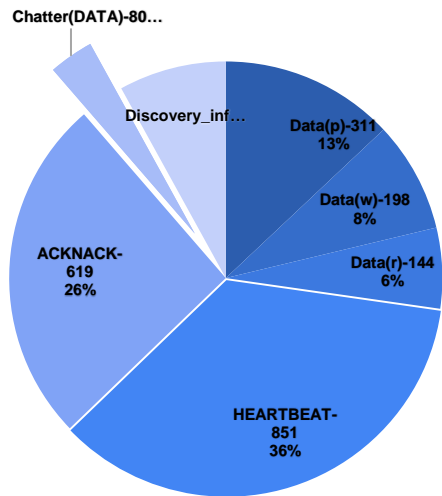
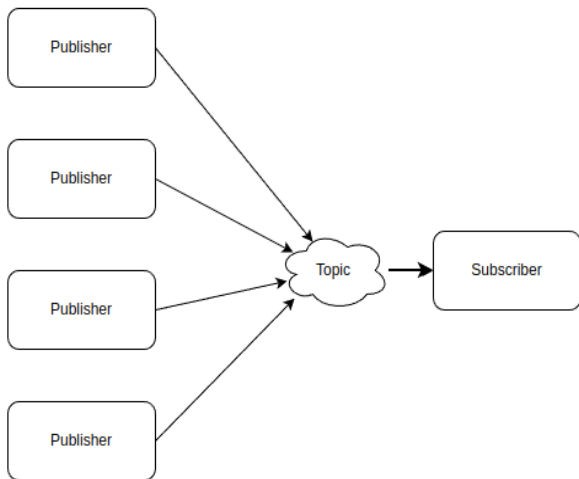
- 전체 패킷량에서 실제 데이터량은 3%에 불과 (5 Node 기준)
- 97%의 패킷은 DDS의 Discovery 과정에서 발생

Analysis of DDS Communication



❖ DDS 통신에서 사용되는 패킷 종류 분석

- 전체 패킷량에서 실제 데이터량은 3%에 불과 (5 Node 기준)
- 97%의 패킷은 DDS의 **Discovery** 과정에서 발생

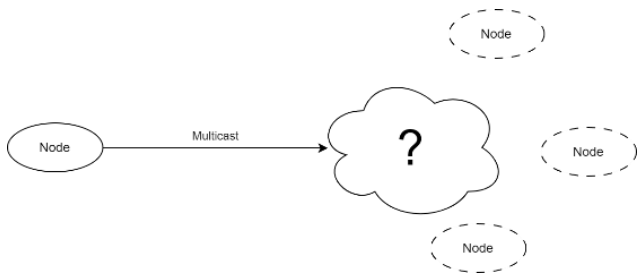


Discovery

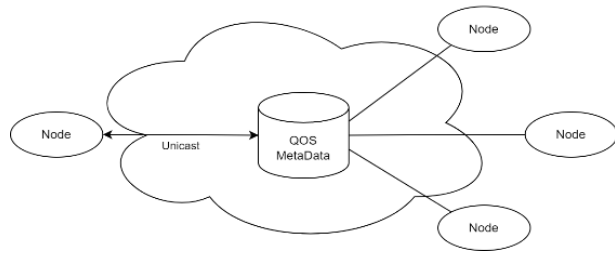


❖ Discovery

- DDS는 네트워크 설정 없이 토픽 이름을 기준으로 연결 가능
- 노드 간 통신을 위한 절차로 동적 노드 확인을 위한 절차 필요
- PDP(Participant Discovery Protocol) : 다른 노드를 찾기 위한 방법
- EDP(Endpoint Discovery Protocol): 찾은 노드의 정보를 공유하는 방법



[PDP]



[EDP]

Discovery



❖ Simple Discovery

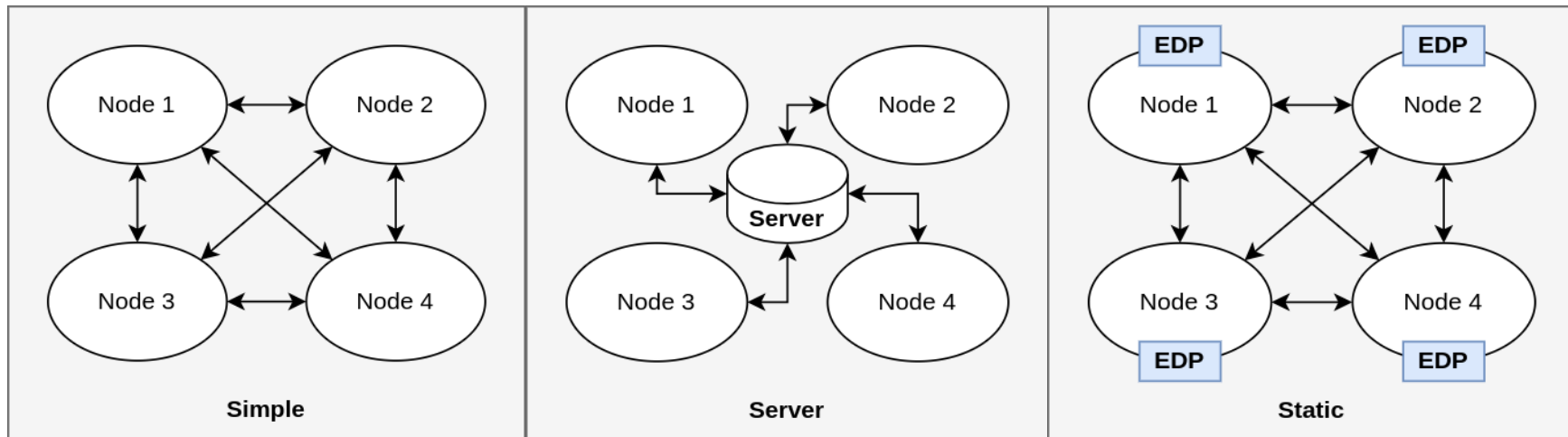
- DDS 에서 기본으로 활용되는 메커니즘 (기본적인PDP와 EDP를 모두 수행하는 Discovery)

❖ Server Discovery

- PDP 서버 (Master)를 활용하는 메커니즘

❖ Static Discovery

- EDP를 노드 내부에 저장, PDP만 수행하는 메커니즘

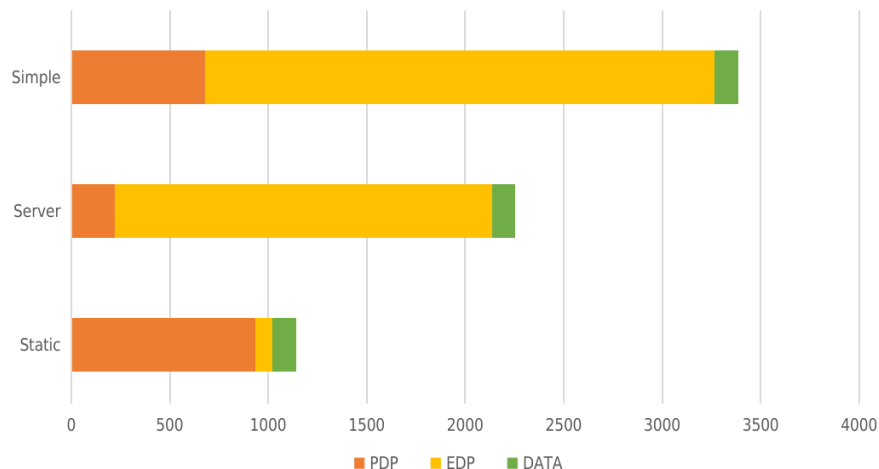


Analysis of Discovery in DDS Communication



❖ 패킷 발생량 측정

- Static Discovery의 경우 Simple과 비교하여 EDP 과정 통신량이 급격히 감소
- 다만, 정의된 EDP 파일의 변화를 인식하여 PDP로 데이터의 변화를 알리는 과정에서 PDP 증가



Proposed Discovery Mechanism

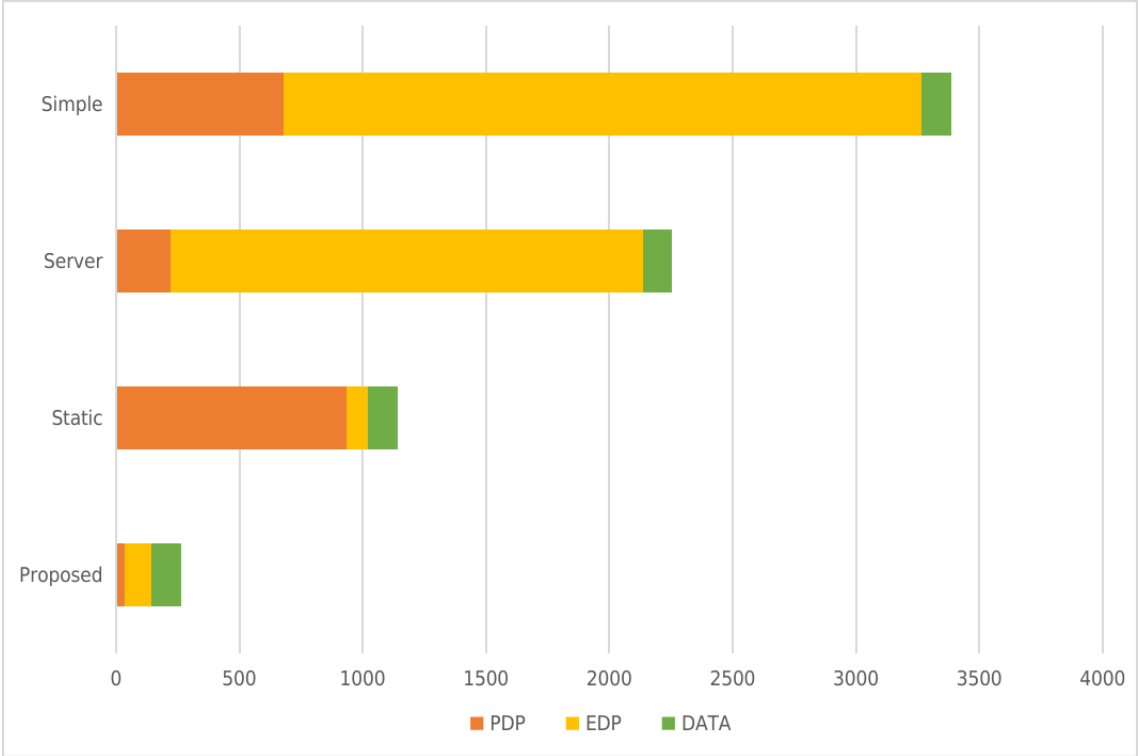


❖ 전달하는 정보가 사전에 정의된 다수의 무인 이동체 운영에 적합한

Discovery 방식 제안

- 송수신하는 토픽 정보는 사전에 정의되었다고 가정 → EDP 불필요
 - Simple Discovery → Static Discovery
 - 사전에 정의된 Topic이 포함된 EDP 정보를 임무 수행 전 각 노드에 설정함으로써 EDP를 위한 데이터 전송 제거
- PDP 과정 간소화
 - Built-in Property 정보가 추가될 때 마다 정보를 전송하는 대신, 주기적인 업데이트를 방식 활용
 - Unicast 대신 Broadcast 방식을 활용함으로써 패킷량 개선

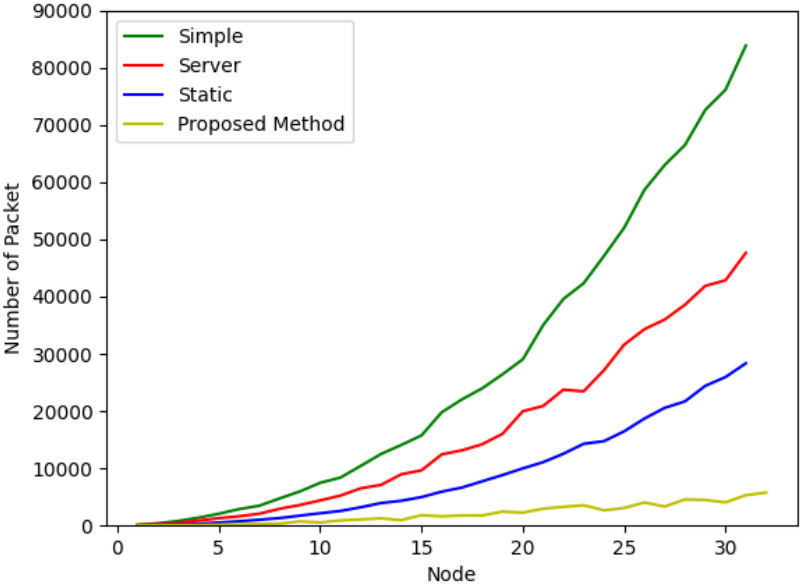
Proposed Discovery Mechanism



Proposed Discovery Mechanism



❖ DDS Discovery 별 패킷 측정 결과



Experimental results on packets according to discovery method of Fast DDS, Including the proposed method

Discovery Type	Value
<i>Simple</i>	76134
-	- 44%
<i>Server</i>	42800
-	-66%
<i>Static</i>	25943
-	-95%
<i>Proposed Method</i>	4051

Experimental results on packets according to discovery method of Fast DDS, Including the proposed method

05

Conclusion



Conclusion



- ❖ ROS2가 대세입니다.
- ❖ 하지만, 제대로 사용하지 않으면 빛 좋은 개살구 입니다.
- ❖ 그래서, 4세부에서 개발중인 PX4-ROS2는 다수 무인 이동체들 운영에 최적화하였습니다.