목차

[아이디어 2](#_Toc377826328)

[[20140117] 함정 전투체계 DDS에 대한 대안 2](#_Toc377826329)

[[20140118] 네트워크 IO 모델 비교에 대한 접근 2](#_Toc377826330)

[용어 3](#_Toc377826331)

# 아이디어

## [20140117] 함정 전투체계 DDS에 대한 대안

함정 전투체계에서 사용하는 DDS는 사용하기에 복잡하고, 고가이다. 또한 따라서 이에 대한 대안을 제시해 보고자 한다. DDS 장점 중 하나는 데이터 버스 개념이다. 따라서 IP가 없기 때문에 유지 보수에 특히 편리한 점이 있다.

나의 아이디어는 Push + UDP를 이용하여 하나의 릴레이 서버가 모든 메시지를 수신한 후 해당 클라이언트로 전송하는 개념이다.

이를 위해서 이렇게 해도 되는 이유는 사격제원계산에 필요한 시간 이내에 메시지 전달이 완료됨을 보이거나. 이 방법의 장점을 설명하거나, 또 굿이 함정 전투체계가 아닌 다른 무기체계 내지는 사람들이 잘 모르는 거 하나를 슬쩍 끼워 넣는 방법은 어떨까 싶다.

* 이미 표준으로 자리잡은 부분에 대항하는 것은 좀 어리석어 보이지 않을까? 강 선배는 문제가 없다고 하지만 말이다.
* 또 이 방법의 단점은 DDS는 단순히 전송 방법의 문제만이 아니라는 것이다. QoS 관련된 부분까지를 고려해야 하고 그렇다면 너무 많은 분량을 구현 & 실험해야 한다.

## [20140118] 네트워크 IO 모델 비교에 대한 접근

연동단 몇 개에 대하여 proactor, thread per connection 을 비교하는 것은 학술 논문으로서의 가치가 조금 떨어져 보인다. 왜냐면 성능 차이가 두드러지지 않을 것 같기 때문이다.

따라서 전투체계 DDS에 대한 대안으로서의 기본 성능 평가 관점으로 접근하자(그래 봐야 클라이언트 개 수가 100여 개로 늘어나는 차이점밖에는 없지만)

## [2014-02-07] 기존 연구와 차이점 도출

“전투체계 시스템을 위한 실시간 환경에서의 비동기 이중화 기법 연구” 논문을 보면 다음과 같은 특징을 갖는다.

* Standby 노드가 Hot-standby로 운용되지 않는다. 때문인지는 모르겠으나 그럼으로 인해서 동기화 시켜야 할 이중화 데이터라는 것이 별도로 존재한다. 한석 버전은 hot standby라 동기화 시켜야 할 별도의 이중화 데이터라는 것이 없다. 아니다 failover가 후 죽었던 active가 standby로 실행되어야 하는 경우 active와 상태를 동기화 시켜야 한다.
* 메시지 수신을 라이브러리가 수행한다. 어떻게 메시지를 라이브러리가 수행하게 할 수 있을까?

[아이디어]

* 액티브 노드의 비즈니스 쓰레드는 이중화 쓰레드보다 우선순위가 높게 한다. Hot standby 노드는 이중화 쓰레드가 비즈니스 쓰레드보다 우선순위를 높게 한다. 이유는 각자 할 일의 우선순위가 틀리기 때문이다. 물론 절체failover가 시작되면서부터는 우선순위도 서로 변경되어야 한다.

## [2014-02-18] 테스트 프로그램 성능 matrices

“Evaluating the Performance of Pub/Sub Platforms for Tactical Information Management” 자료를 보면 benchmark는 2 가지 항목이다. Jitter 항목은 단지 Latency의 차이를 보이기 위한 자료 정도.

* Latency: 메시지 송신 후 수신 ACK를 받을 때 까지 시간

송신 후 시간 기록 & AckMsg 수신 후 시간 기록하여 계산한다.

* Throughput: 주어진 시간 동안 보낸 메시지의 수

Timer를 이용하여 time-out 이벤트가 발생하면 송수신 중지

* Jitter

## Latency 측정 방법

|  |
| --- |
| const short MAX\_MSG\_LENGTH = 16384;  struct PubMsg {  long id; /\* MAX\_MSG\_LENGTH에 따라 변동 \*/  long size; /\* MAX\_MSG\_LENGTH에 따라 변동 \*/  long seqnum;  byte data[MAX\_MSG\_LENGTH]; // 4, 8, 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, 8192, 16384 순으로  };  Struct AckMsg {  long id; /\* = always 1 \*/  long size; /\* = always 4 \*/  long seqnum;  }; |

## Throughput 측정 방법

근데 이미 iperf와 같은 성능 측정 도구가 넘쳐나는데 이게 의미가 있나? 아님 iperf를 vxworks에 포팅을 해서 테스트라도 해야하나? 포팅이야 어렵지는 않지만 의미가 있으려면 어떻게 해야하지?

# 용어

STUN->~~RFC3489~~->RFC5389

릴레이 -> RFC5766

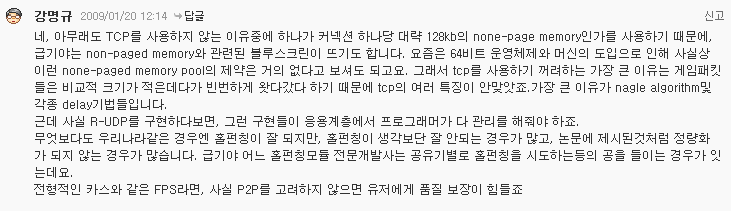
턴서버, 턴클라이언트

TURN server open source: http://turnserver.sourceforge.net/

릴레이(relay)라는 용어를 턴(turn)과 함께 쓴다. 출처 http://cafe.naver.com/ongameserver/5844

TURN-TCP -> RFC6062

P2P로 클라이언트끼리 통신이 불가할 경우 P->server->p 요렇게 통신을 하게된다. 요게 패킷 릴레이라는 개념인가 보다.



-> 즉, TCP는 접속자 수에 따라 부하가 늘어나게 된다 (출처: http://cafe.naver.com/ongameserver/3538)

R-UDP(Reliable User Datagram Protocol): http://en.wikipedia.org/wiki/Reliable\_User\_Datagram\_Protocol -> IETF에 아직 표준으로 정해지지 않은 듯 싶다. 문서가 검색이 안 된다.