RogueQD

McMaster Engineering Competition 2016

Ian Prins

Mikhail Andrenkov

Ori Almog

(Team 8)

What is RogueQD

RogueQD is a communications system for boats and buoys in the deep sea.

Employees the 3 - 3,000 Khz VLF/LF/MF radio band as a medium of communication.

Allows for distress signals to be sent by boats and propagated al about the system.

Simulates transmission error.

RogueQD – Rogue Query & Distress

Basics

Boats can talk to one another (Similar to how walkie-talkies work). Boats can listen on a frequency and broadcast to a frequency. This kind of communicatin is called jargon.

Boats can also send distress signals to buoys. This is done on the 500 Khz channel. See: Distress.

Buoys report weather statuses. Nearby boats that can hear the buoy make note of its calls.

Messaging

Messages are sent by boats and buoys.

Buoys have really good receivers, with high reception strengths. They're built with purpose!

Boats have poor receivers – receivers are big and heavy, and boats tend to be very light.

Boats and buoys have similar transmitters. Transmitters are fairly lightweight, so boats can carry trasmitters that can send a distant signal.

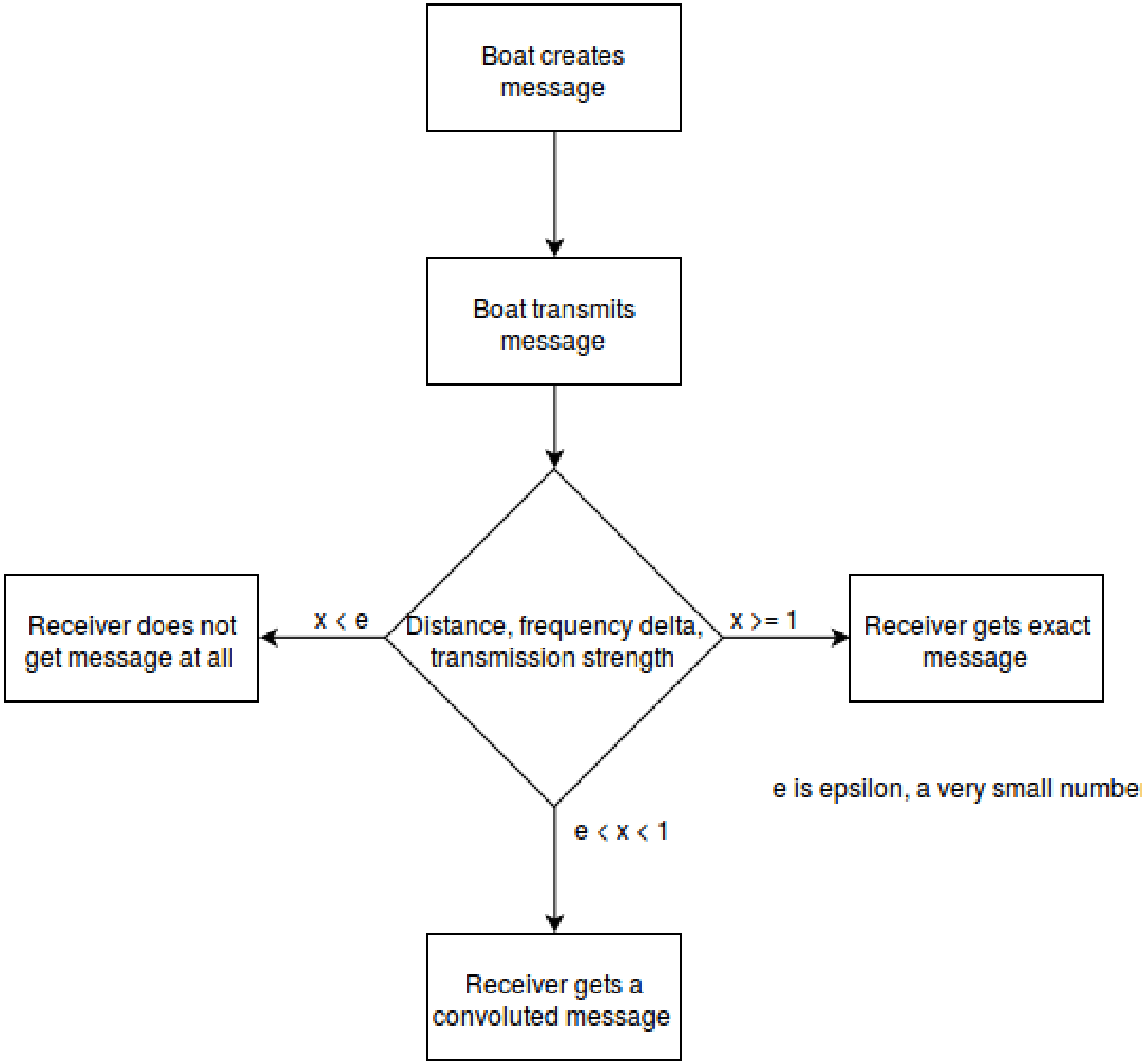
Message reception

Whether or not a message is successfully transmitted depends on:

- Sender transmission strength
- Receiver reception strength
- Distance between the two
- Frequency difference between the two

$$p(M) = \frac{c S_b S_r e^{\left(\frac{-(f_b - f_r)^2}{2\sigma^2}\right)}}{\sqrt{d 2\pi\sigma^2}}$$

Message reception



Boat creates message

Boat transmits message

Distance, frequency delta, transmission strength

x < e

Receiver does not get message at all

x >= 1

Receiver gets exact message

e < x < 1

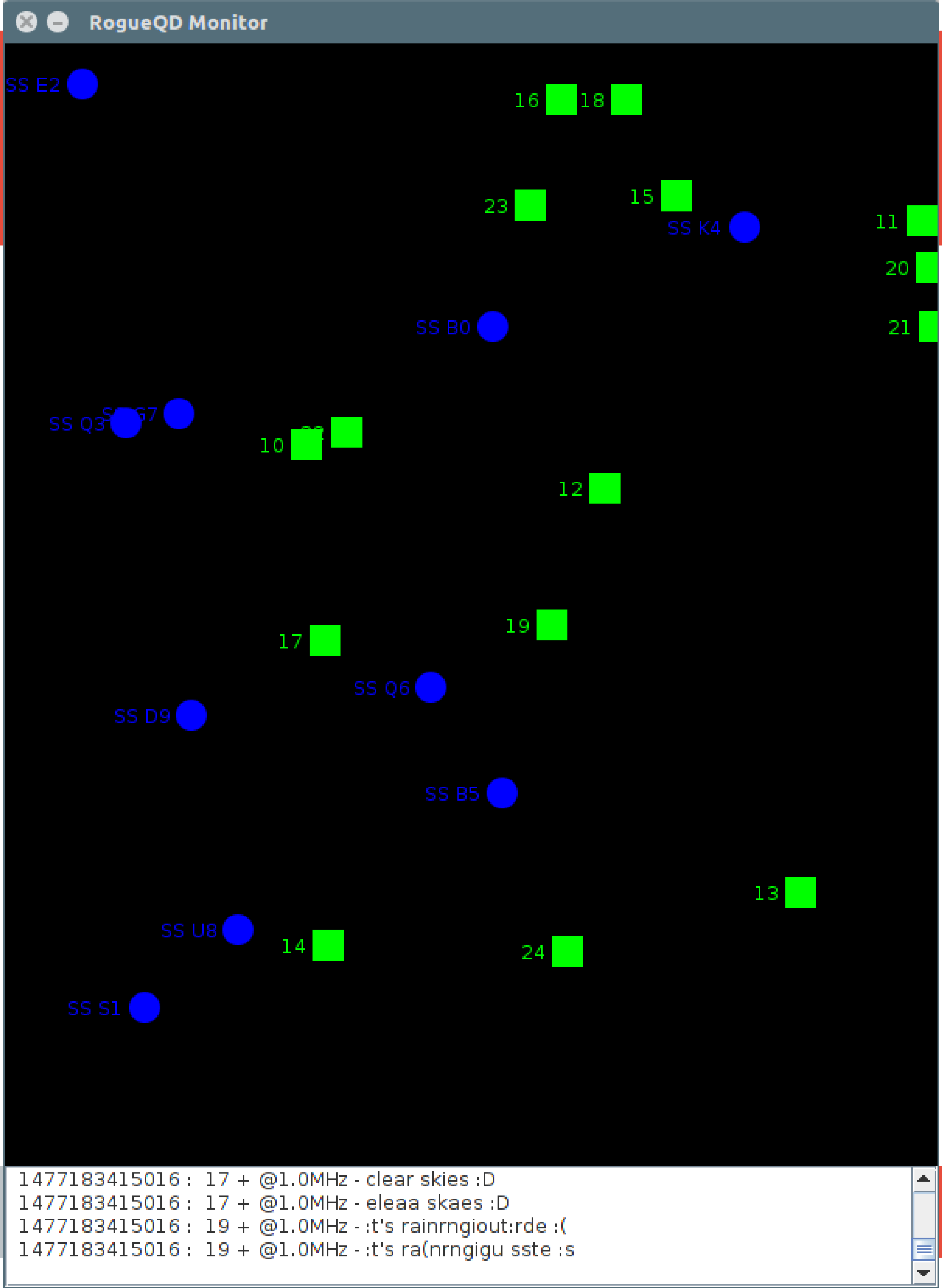
Receiver gets a convoluted message

e is epsilon, a very small numbe

GUI

The technology that was used for the GUI is Java Swing.

- Updates at 60 FPS (Frames per second)
- Vibrant colors
- Green squares: Buoys
- Blue circles: Boats
- Each has a visible ID
- Messages appear at the bottom



```
1477183415016 :  17 + @1.0MHz - clear skies :D
1477183415016 :  17 + @1.0MHz - eleaa skaes :D
1477183415016 :  19 + @1.0MHz - :t's rainrngiout:rde :(
1477183415016 :  19 + @1.0MHz - :t's ra(nrngigu sste :s
```

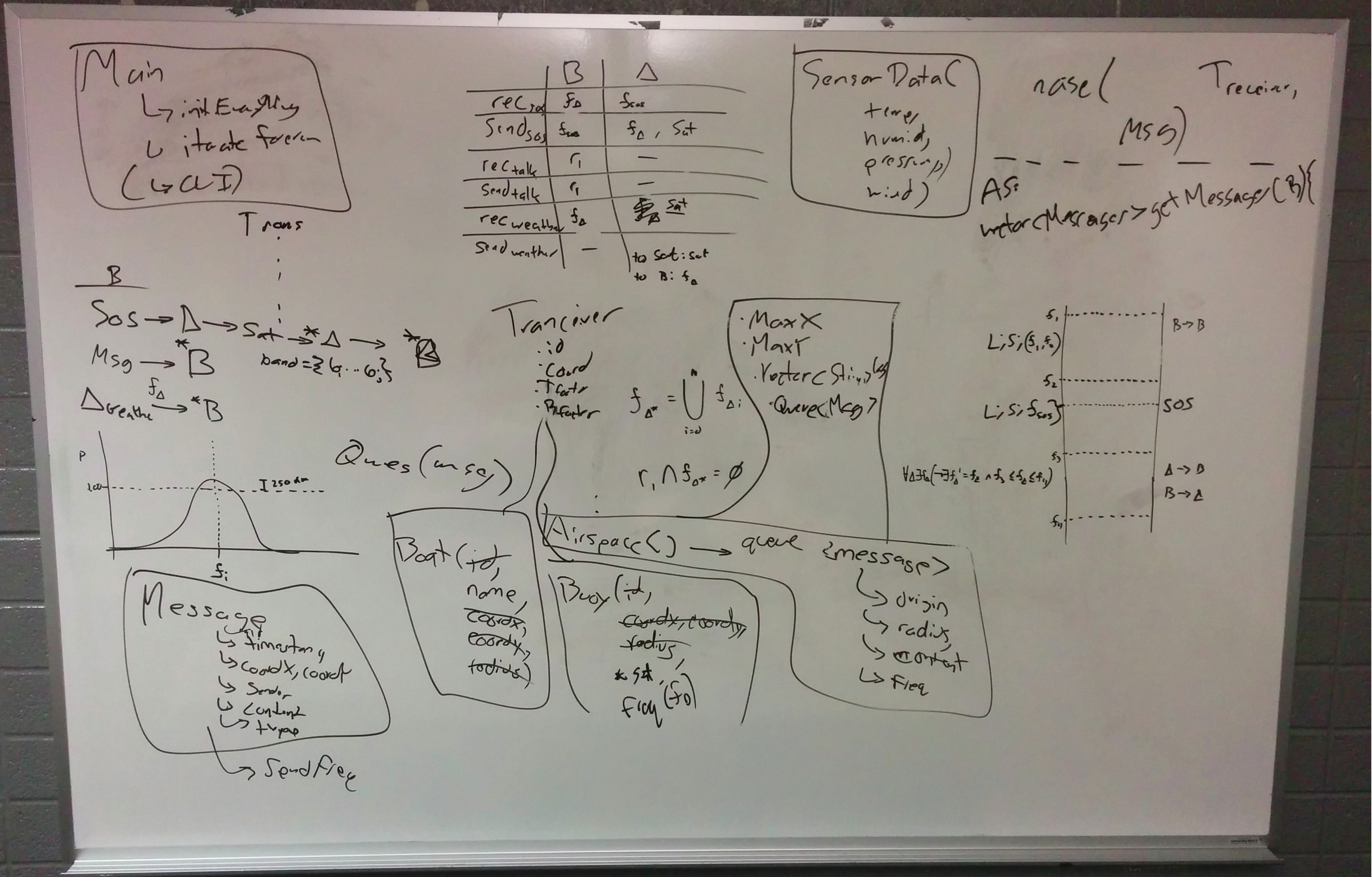
About us

Ian “GOTO >>” **Prins**

Mikhail “Heuristic” **Andrenkov**

Ori “XOR” **Almog**

FSP enthusiasts and third year Software Engineering students



Main
↳ init Everything
↳ iterate forever
(↳ GUI)

Trans

	B	Δ
rec_{sos}	f_Δ	f_{sos}
$Send_{sos}$	f_{sos}	f_Δ , Sat
rec_{talk}	r_1	—
$Send_{talk}$	r_1	—
$rec_{weather}$	f_Δ	Sat
$Send_{weather}$	—	to Sat: sat to B: f_Δ

SensorData(
time,
humid,
pressure,
wind)

raise($T_{receiver}$,
Msg)

AS:
vector<Messages> getMessages(B){

B

$Sos \rightarrow \Delta \rightarrow Sat \rightarrow {}^*\Delta \rightarrow {}^*B$

$Msg \rightarrow {}^*B$

band = $\{6_1 \ldots 6_5\}$

$\Delta_{weather} \xrightarrow{f_\Delta} {}^*B$

Tranciver
· id
· Coord
· Tfactor
· Rfactor

$f_{\Delta^*} = \bigcup_{i=0}^{n} f_{\Delta i}$

$r_1 \cap f_{\Delta^*} = \emptyset$

· MaxX
· MaxY
· Vector<String> ids
· Queue<Msg>

f_1

$L_i S_i (f_1, f_2)$

B→B

f_2

$L_i S_i f_{sos}$

SOS

f_3

$\forall \Delta \exists f_\Delta (\neg \exists f_\Delta' = f_2 \wedge f_3 \leq f_2 \leq f_{11})$

Δ→B

B→Δ

f_{11}

Ques(msg)

Message
↳ timestamp,
↳ coordX, coordY
↳ Sender,
↳ Content
↳ type

↳ SendFreq

Boat(~~id~~,
name,
~~coordX~~,
~~coordY~~,
~~radius~~)

Airspace() → queue <message>
↳ origin
↳ radius,
↳ content
↳ Freq

Buoy(id,
~~coordx, coordy~~,
~~radius~~,
* Sat,
Freq (f_Δ)