Communication protocol

between anchors and token

1. Summary

This document covers the communication protocol between the anchors and the tokens that enables them to exchange information and commands. Initially this document’s scope was limited to the token distance measurement strategies in an anchor environment, but it has been extended to cover a variety of commands and information exchanges.

The distance measurement strategy is activated through a command directive from the anchor, that will be further discussed in this document.

The distance measurement strategies can be 2-folded.

- Measure directly from token to token as is done in the crowdband implementation.
  This implementation results in an immediate distance measurements, because it is executed on tokens itself.

- Measure distance from multiples anchor to same token, such that with triangulation the location can be determined. After having all the locations determined on all the tokens, the distances between the tokens can be deduced. This strategy involve a complete overview of tokens the area, before any real calculations can be performed.

Before the distance measurements can take place, other preparations need to be taken care of, which will be discussed in this document as well.

2. Overview

This overview uses an office space as an example environment to explain the components that comprise of a presence detection system. There are two components:

a. Anchor that detects the presence of any tokens. Anchor consists of a
   - connection to a central station (WIFI/Lora)
   - SUM4 module with UWB chip
b. Token: a SUM4 module that beacons a presence signal periodically

In an office space several anchors are placed at several locations, such that the entire office space is covered with the UWB range of all the anchors. In locations where no UWB coverage is possible or desired, no presence detection can be accomplished.

Tokens that wander in the office space are detected by anchors and this presence information is uploaded to the central station for further processing. Please see section "Data format of presence information" for more details.

Communication between anchor and token is initiated by the token. The token beacons a presence message and if an anchor detects a presence message, automatically an acknowledge message is sent back from the anchor that may contain control information upon which the token should subsequently act.

This mechanism within the presence detection system, allows the anchors to activate functionality on tokens if that is desired. This can be done on an individual token or group(s) of tokens or all tokens that are detected. The anchors will receive control information for the tokens from the central station. In case an anchor detects the presence of designated token, the control information can be forwarded to the token during the ACK response.

3. Presence message and ACK message

The token sends out a burst of presence messages periodically. This burst is usually 2 or 3 identical messages aside from the attempt number) that are sent out in rapid succession. The goal with this burst is to reduce the impact of any potential interference and or temporary blocking of the LOS for the signal to propagate.

A presence message contains presence information and may contain an additional request (cmd) to obtain (control) information from the anchor. This request is added to the presence message, should the token detect that its control information has expired or is out dated.

To ensure all tokens have het latest control information a control jacket number is used. The control jacket number is a number that represents the latest issued information from the anchors. We assume here that all anchors are in constant contact with the central station and that they can receive new control information from the central station at any point in time.

On reception of a presence message, the anchor will always respond with an ACK message.

The token decides whether it is going to make an effort to receive an ACK message from the anchor. For instance on every 5-th presence message, the token may decide to actually receive an ACK message from an anchor, while the other times the token doesn't even bother to turn the receiver on.
Please note that an ACK message can only be received when the token and anchor are in UWB range of each other.

The presence message (from token to anchor) has the following format:

Field name	#Bytes	Remarks
UWB packet type	1	Discovery (value 2)
Attempt nr	1	
TokenId	4	
Seqnr	1	
Battery percentage level	1	

The ACK message (from anchor to token) has the following data format:

Field name	#Bytes	Remarks
Anchor Id	1	TokenId of anchor
Control jacket number	1	
Ad-hoc command	1	Ad-hoc command for all tokens
Diagnose command (map)	1	Diagnose command for all tokens
Sync refresh time in ms	2	Time remaining till sync is emitted
Sync window time in ms	1	Length of window in which sync signal is emitted

The ad-hoc command in the ACK message is designated for all tokens. The command itself may be enough of an indication for the token to proceed, however it can also be foreseen that the token needs to download control information (see below) from the anchor before it is able to execute the command.

The diagnose command in the ACK message is a method to enable some diagnostics on the token. This can range from flashing leds to requesting to send more information (measurement results) to the anchor, or storing more information on the token on every cycle.

The SYNC refresh time and SYNC window time indicate the time point and time-range at which a SYNC signal will be emitted. As this information is part of an ACK, the contents of the ACK message will be constantly updated by the anchor to reflect the upcoming sync time, the update to the SYNC refresh time occurs in the ballpark of 10 times per second.

Automatic ACK message contains predetermined information

The anchor emits and ACK message when it receives a presence message. This ACK message is pre-programmed ahead of time, so when the anchor auto-replies it is done with the pre-programmed information. This limits the usefulness of the ACK message as it can’t be applied to a specific set of token-ids.

A token-specific command will be added to the design that resolves this issue.

4. Command Control information

In contract to the ad-hoc command in the ACK message is the control information. The control information provides more details on the command execution, which tokens are involved and when the execution will take place. For instance some commands may require coordination between multiple tokens.

The control information (from anchor to token) has the following data format:

Field name		#Bytes	Remarks
Control jacket number		1	
Control jacket version number		1	Version 1
Number of TokenIds M		1	
Number of TokenId ranges N		1	
Number of Commands R		1	Numbers of commands to execute
M times	TokenId 1	4	
N times	TokenIdRange 1 min	4	
	TokenIdRange 1 max	4	
R times	Cmd 1	1	1: distance to anchor 2: distance to tokens (anchors included) 3: flash led with color as indicated in parameter field
	Iterations 1	1	Number of times cmd to execute
	Time 1	1	Relative time in ms, cmd to execute
	Parameter 1	1	

The control information has a more permanent character than the ad-hoc command. Because of the size of the message, this information needs to be requested by the token to avoid an unnecessary load on the ether when transmitted constantly.

5. Communication between anchors and tokens

The communication in a presence detection system is very simple in its implementation, tokens emit a presence signal at will and anchors receive (detect) those signals. The anchors are continuously receiving to enable receiving presence messages at random at all times.

This basic form of communication is adhering to the following guidelines.

- Efficient use of aether
- Efficient use of power, where the balance of power consumption will always fall in favour of the token

If the presence detection system is augmented to allow for other types of functionalities, the communication capabilities between the anchors and tokens need to be updated and adhere of course to the guidelines for efficiency and power consumption.

The simplest way of augmenting communication capabilities is to make use of a SYNC signal, also referred to as a pilot (signal). The ideal form is that a SYNC signal (pilot) is emitted periodically and is received by all the anchors and tokens, such that future actions can be coordinated in relation to this SYNC signal.

Unfortunately the availability of a SYNC signal is more complex in nature. The radios have limited range, require too much power, or the propagation of a SYNC signal through several entities causing other (technical) challenges. The availability of a SYNC signal is not an absolute must, but it makes the coordination of flow of messages easier to design and understand and reduces the possibilities for interference.

Without going into much detail, I am highlighting several approaches to augment the communication capabilities. Only one of these approaches will be chosen, while the other approaches serve as a place holder for future development.

The approaches to augment communication capabilities are:

- Time of presence message based with relative offset for future actions
- Time of SYNC message that is emitted and unique per anchor
- Time of SYNC message that is synchronized across all anchors.

The approach taken is the time of SYNC message that is emitted per anchor. This approach doesn't require all anchors to be synchronized and work either way.

6. Augment of communication capabilities

The augmentation is focused on the GGD project, so it won't be all inclusive.

6.1 Tokens

The following directives are sent from token to anchor:

- Request sync information
- Request control information
- Request data upload to anchor

6.1.1 Request sync information

A token wanders around and every time its presence is detected by an anchor, it can request the SYNC information from the anchor, provided the token needs the SYNC information to participate in any activities other than presence detection.

Field name	#Bytes	Remarks
Command	1	CMD_REQ_SYNC_INFO
Anchor Id	1	
TimeOffset (ms)	2	>0 : when response is expected from anchor

6.1.2 Request control information

A token has the ability to request control information regarding command execution. It must request this when the control jacket number is expired. It is assumed that all anchors will always have the latest control information available.

Field name	#Bytes	Remarks
Command	1	CMD_REQ_CONTROL_INFO
Anchor Id	1	
TimeOffset (ms)	2	>0: TimeOffset when response is expected from anchor after the presence messages have been sent (50 ms time window) 0: default 100 ms after SYNC of anchor id

6.1.3 Request data upload to anchor

A token can request a data upload to the anchor at any time. It usually will be done at the end of a test, but can also be requested when the token detects that the battery power is getting depleted.

Field name	#Bytes	Remarks
Command	1	CMD_REQ_DATA_UPLOAD
Anchor Id	1	
TimeOffset (ms)	2	>0: TimeOffset when response is expected from anchor after the presence messages have been sent (50 ms time window) 0: default 100 ms after SYNC of anchor id

6.2 Anchors

The following directives are sent within the ACK from the anchor to token:

- Initiate distance measurements between tokens
- Initiate distance measurement to anchors
- Token-specific command
- Idle command
- Flash led red (diagnostic)
- Flash led green (diagnostic)
- Flash led blue (diagnostic)

6.2.1 Initiate distance measurement between tokens

The distance measurement process is initiated on the token, its execution details are taken from the command control information in case this is up-to-date. Otherwise default execution details are assumed for this process.

6.2.2 Initiate distance measurement to anchors

The distance measurement process is initiated on the token. Its execution details are taken from the command control information in case this is up-to-date. Otherwise default execution details are assumed for this process.

6.2.3 Token-specific directive

The anchor indicates that there is a token-specific directive that may be sent to the token

- directly after the SYNC message has been sent out, in case the anchor has SYNC signal capability activated
- during a 40 ms window after the last iteration of the presence message is received.

The anchor will only send this token-specific message out, if the presence of this token has been detected.

6.2.4 Idle directive

When token receives the ACK message with this idle directive, it will disable the execution of any pending commands and just reverts back to the emission of presence messages.

6.2.5 Flash led (red, green, blue) (diagnostic)

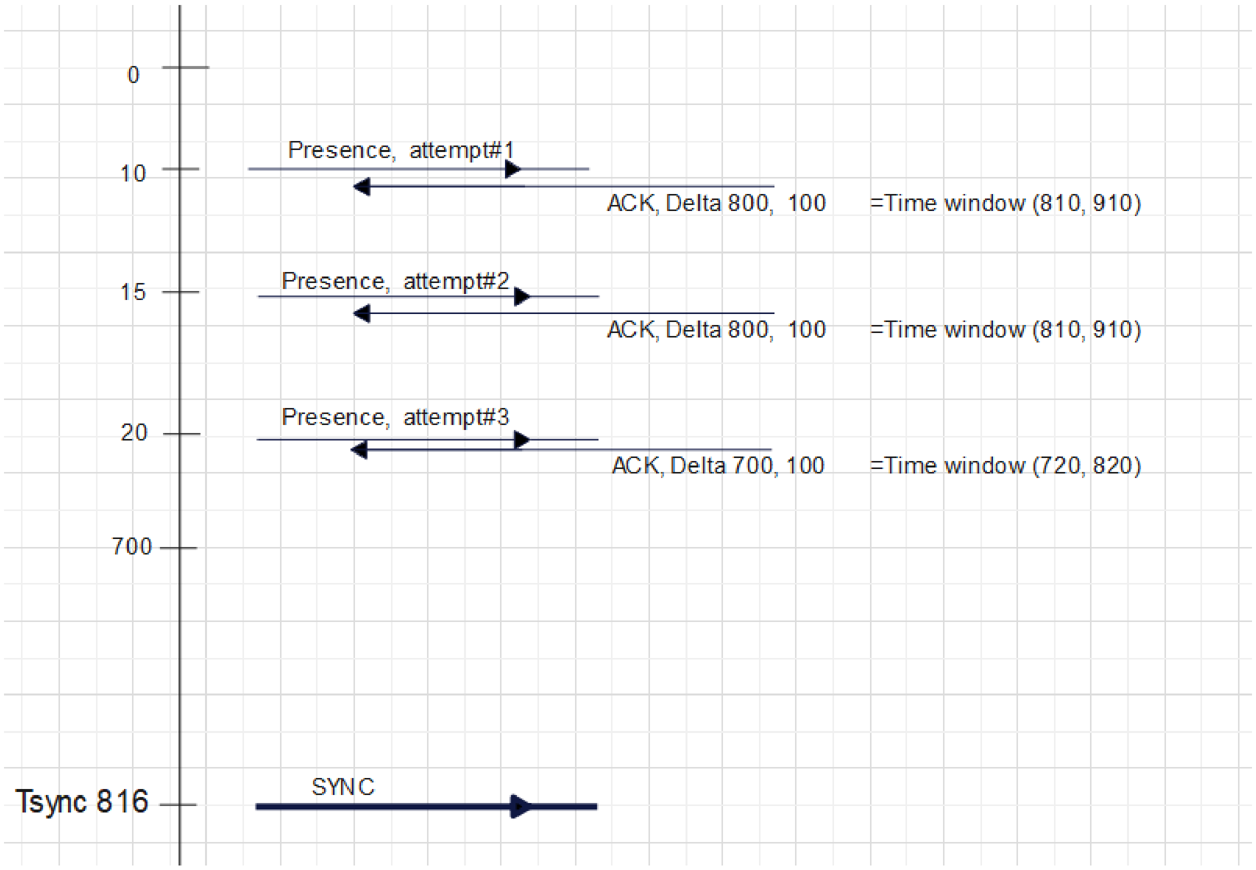
When token receives the ACK message with the flash led directive, it will flash the led for a short period of time.

7. Execution of commands

The following diagrams are

- Obtain sync information
- Request control information
- Request data upload to anchor

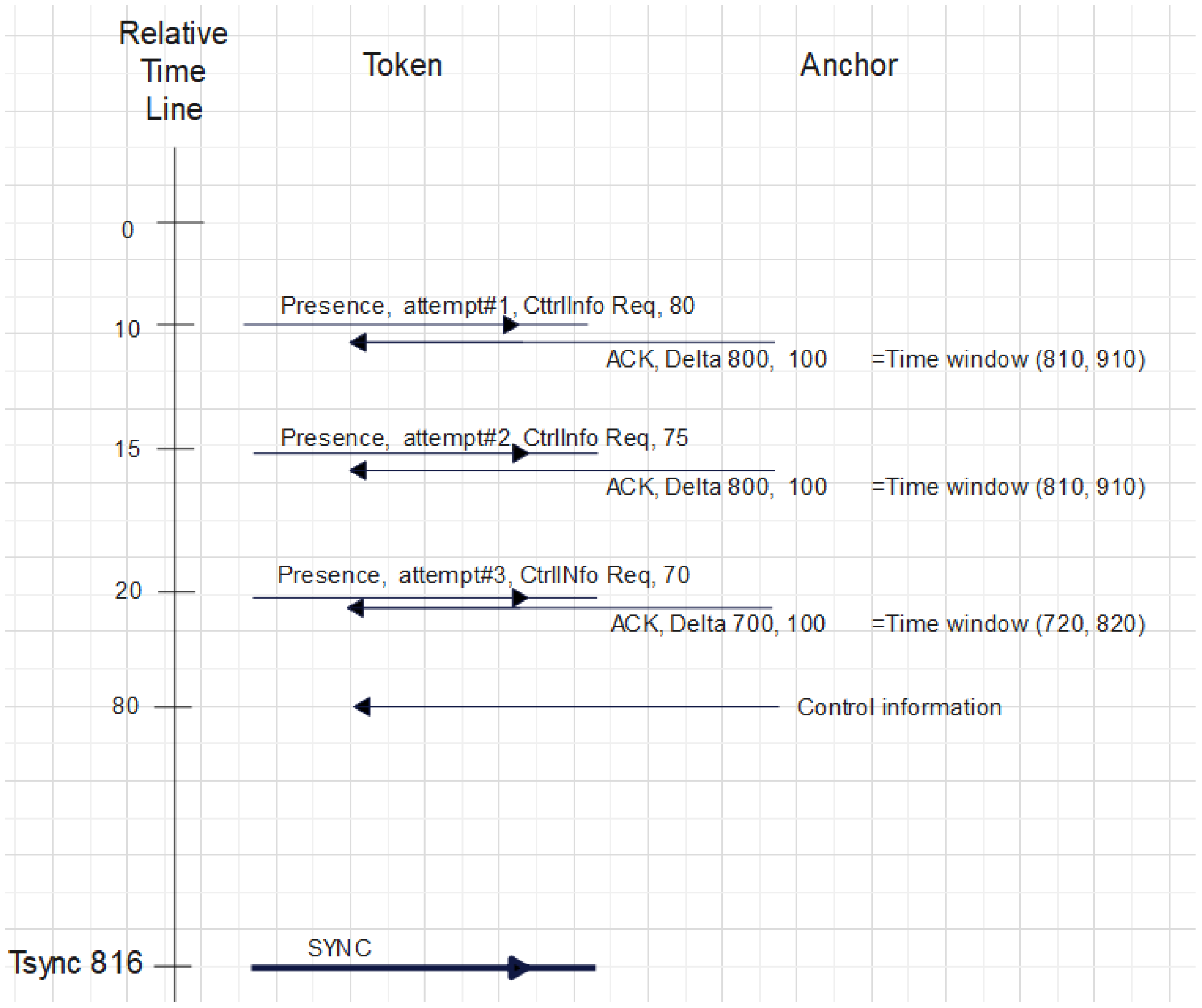
7.1 SYNC info always sent with the ACK (anchor continuously updating)



This diagram shows how the token emits presence messages that are received by the anchor. Once the token has seen the SYNC message it may adjust the presence message schedule so that it won't interfere with other tasks that it is required to do (for instance distance measurements).

The anchor constantly adjusts the contents of the ACK message to reflect the time window in which the SYNC signal will be sent out.

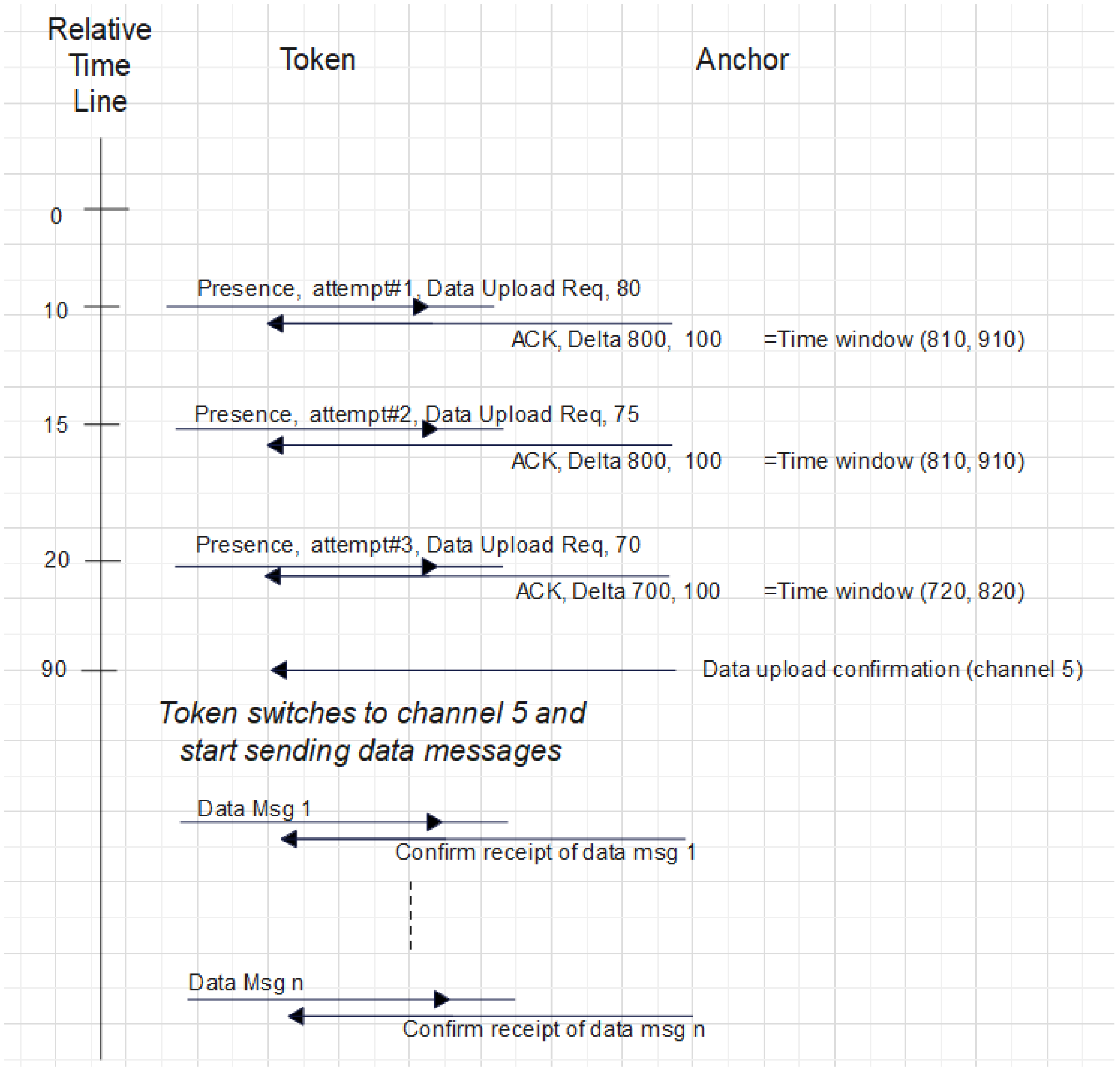
7.2 Request control information



The token sends the control info request to the anchor and expects a response 80, and then 75 and then 70 ms after the anchor receives the presence messages.

The anchor might receive one or more presence messages, but they all indicate the same time the anchor must respond to the tokenId. It is unsure whether the answer should be sent out on a different channel, (token must also indicate that then as well) in order to reduce the interference potential.

7.3 Request data upload


Relative Time Line
Token
Anchor
0
10
15
20
90
Presence, attempt#1, Data Upload Req, 80
ACK, Delta 800, 100
=Time window (810, 910)
Presence, attempt#2, Data Upload Req, 75
ACK, Delta 800, 100
=Time window (810, 910)
Presence, attempt#3, Data Upload Req, 70
ACK, Delta 700, 100
=Time window (720, 820)
Data upload confirmation (channel 5)
Token switches to channel 5 and start sending data messages
Data Msg 1
Confirm receipt of data msg 1
Data Msg n
Confirm receipt of data msg n

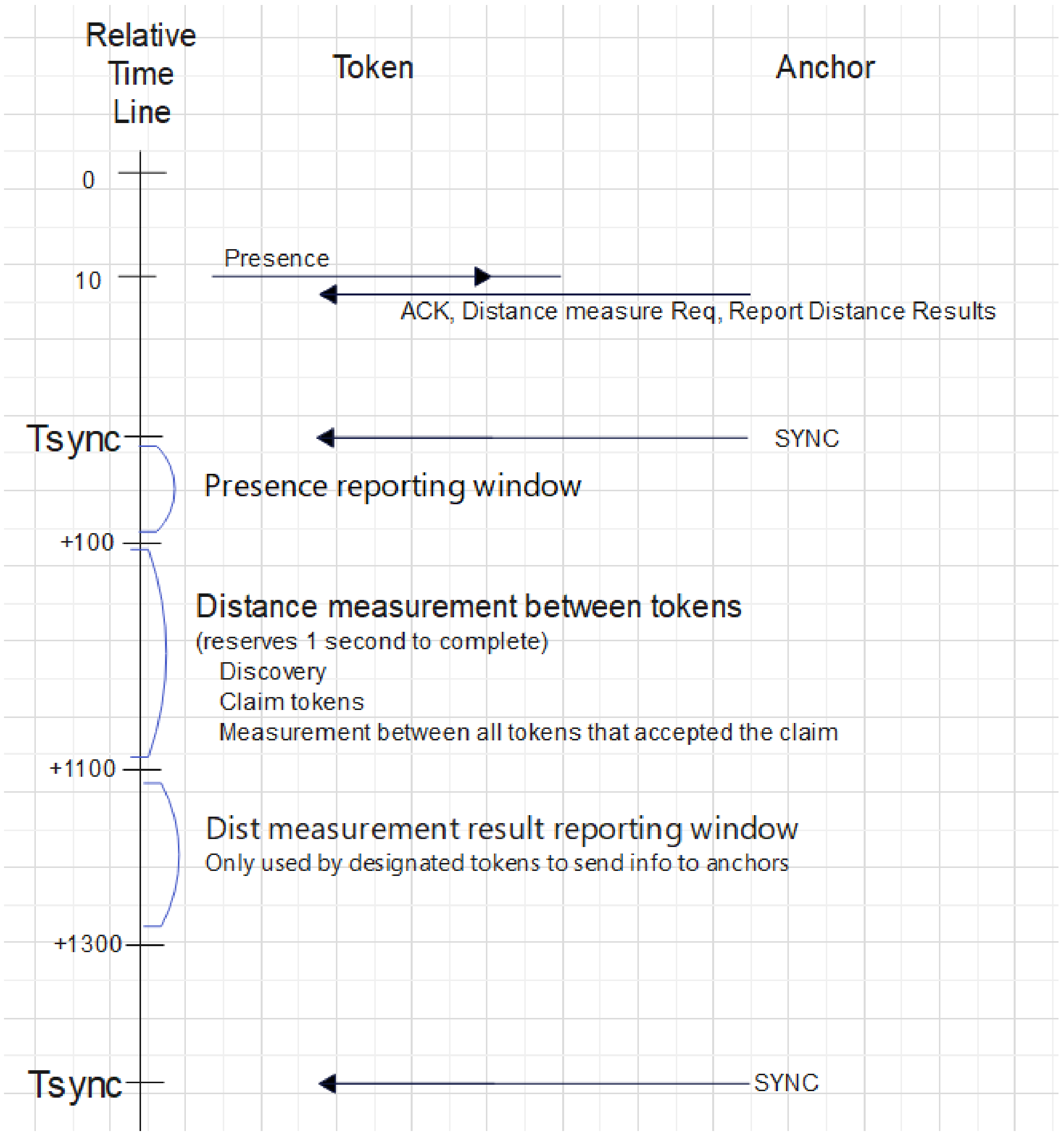
The token sends a request out for data upload and waits for the anchor to response at time point 90 as indicated.

If the anchor is willing to receive the data upload, the anchor will respond at time point 90, with the confirmation to the token to start uploading the data. Every chunk of data that gets uploaded is acknowledged in this figure, however it may be implemented in a more efficient manner.

After the token has uploaded all data and received confirmation from the anchor that all data has been received, the token clears its memory and continues with logging of information.

The anchor will be temporarily out of commission to detect presence messages while taking care of the upload. Once the upload is finished, the anchor resumes its presence detection and other regular actions.

7.4 Initiate distance measurements between tokens


Relative Time Line
Token
Anchor
0
10
Presence
ACK, Distance measure Req, Report Distance Results
Tsync
SYNC
Presence reporting window
+100
Distance measurement between tokens
(reserves 1 second to complete)
Discovery
Claim tokens
Measurement between all tokens that accepted the claim
+1100
Dist measurement result reporting window
Only used by designated tokens to send info to anchors
+1300
Tsync
SYNC

In the picture above, the token that receives an ACK indication distance measurement, will shift in the future its presence messages reporting to the first 100 ms time window after the SYNC.

The token will then commence the distance measurement strategy as used in the crowdband, the anchor may participate in this strategy as well, still undecided whether this is desired.

In case the token received also the directive to report the distance measurement results, the token selects a random time within the distance measurement reporting window to report the result to the anchor.

7.5 Initiate distance measurements to anchors (for anchor triangulation)
7.6 Token-specific command
7.7 Flash led (diagnostics)
7.8 Report distance measurement results

More to come………………………..

Anchor/Gateway interface

Format of presence information sent from anchor UWB to gateway (raspberry/lora)

Different formats for different information:

V1

WIFI: anchorId, seqnr, tok1, seqNr , bat, rssi tok2, seqNr, bat, rssi, tok3, seqNr, bat, rssi, tok4, seqNr,, bat, rssi, tok5, seqNr, bat, rssi,

LORA: anchorId, seqnr, tok1, seqNr, tok2, seqNr, tok3, seqNr, tok4, seqNr, tok5, seqNr,

V2

TBD