Design for Internet of Things

Prof. T. V Prabhakar
Department of Electronic Systems Engineering
Indian Institute of Science, Bengaluru

Lecture - 11
RFID Theory - 02

Alright, so let us understand this anti coalition part of RFID, along with sessions, there is also something called sessions, which is part of the RFID EPC Gen two protocol, which you should know, for the following reason that often you are not talking about one reader being used for detection of tags, but in fact, there will be multiple readers to ensure that any time the tag has is depolarized, the other reader can read, this is one reason why you may need to readers.

Second reason is think about a smart shelf, go to a department store, you see a number of items in this shelf, you put one reader for each shelf, let us say, and this reader is continuously keeping track of each item in the rack, actually as an RFID tag. If there are, let us say, 1000 items in that rack, there are 1000 tags. And then the reader is actually continuously reading inventorying all these tags, which are corresponding to those items.

It is very important because if you remove one, customer comes and removes one item, you now know the count very quickly. So, this is reason why you may want to put one reader it is also possible that the person at the cash counter, or the place where the building happens, he has another reader, and a customer asked a question about availability of an item. So this person, depute, let us say somebody who is out there to go and check whether that item is actually there on the rack.

This is just giving you information continuously despoiling, but then you also want to go and check manually. I mean, manually when I say a human goes there with another reader, and then wants to check for that particular item. In that situation, you are ending up with a population of tags, but there are two readers, one that is doing something continuously and one that is come with the in the presence of the existing RF field.

There is another reader that has come in. Now question is can two readers coexist? Can they inventory the same population of tags? That is the key point. And how is that possible? Remember, one guiding thing, a tag can be read only one by one reader at a given instant in time. It is not possible that the tag can talk to two readers at the same time, it is not going to work. Two readers cannot say I want information and put RF energy at the same instant in time.

If you think of time along the x axis, a tag response comes only from to one reader at a given instant in time. That means you are to interleave one reader, next time reader 1, then next time reader 2, reader 1, reader 2, reader 1, reader 2, that you have to ensure this is the key point. So do not forget that it has to be time interleaved. If you are talking about multiple readers, how does the tag know, question is this.

There is a state machine there is no controller there I mentioned and how does the tag know whether this is coming from this reader, reader 1 and not from reader 2? That is the beauty. It actually stores another piece of information called related to sessions. Reader 1 can come from one session and reader 2 can come from another session. There is a flag maintained on the RFID tag with respect to each session.

It stores the information related to that session. What is that flag information? Flag information is related to how the tag has to respond to that particular session. That information is held there. By that, just that one flag. We will see all of that as we go along because very important when you talk about multiple readers, but I am sure even before you do all of that you have questions related to understanding this anti-collision part in a little more pictorial manner.

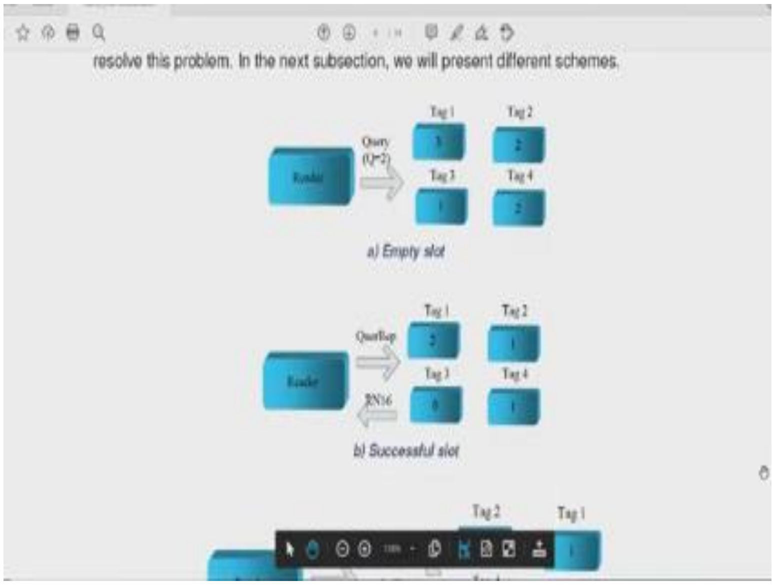
I want to draw your attention to something I saw which you can download yourself and read it thoroughly. So, this is that document here.

(Video Starts: 05:06)

So, this document, essentially is anti-collision issue analysis in generation two protocol. If you come down, you will see some nice pictures. So let me come down directly to the picture of interest. Now, there is some general discussion about different types of anti-collision algorithms.

But we will not worry about to, we will just look at EPC Gen 2. Look at this, these two pictures here.

(Refer Slide Time: 05:30)



resolve this problem. In the next subsection, we will present different schemes.

One of them says Q value that is supplied is two from the reader using the Query command and there are tag 1, tag 2, tag 3 and tag 4. Now, the answer here is empty slot. The reason is, none of them when they threw their two random numbers, one corresponding to RN-16. And the other number corresponding to the right to communicate comes when you have 0, that did not happen here.

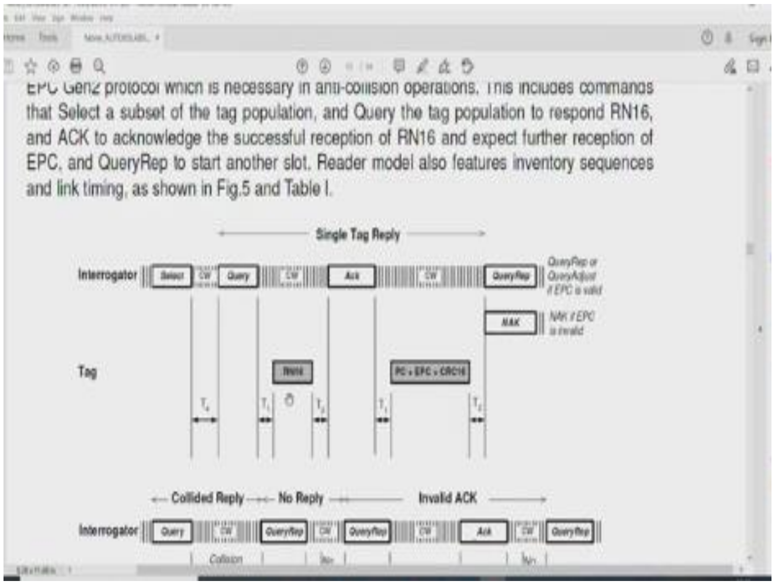
Therefore, what happened? It became an empty slot, it just turned out that none of them chose zero during the second. So, think about it has two coin tosses. First coin toss gives you a random number 16 one random number and the second time the coin toss will give you the slot, right to communicate slot provided it comes to 0. But therefore, it did not come. So, it got an empty slot, this is very clear.

Now look at what happened, the reader did not change the Query, it still was Q equal to two, but then it gave a Query Rep command. When a Query Rep command was issued, look at what actually happened. Each one of these tags counted to decremented their value by 1. So, tag 1 became 2 and tag 2 became 1 and so on. It turned out the tag three earlier was one it became 0 and therefore it responded with its RN-16 value is a beautiful understanding of how Query and Query Rep are being used.

Now, it is also possible that when a Query Rep has issued tag 2 and tag 4 both of them counted to 0 is just another case. Where they both collided and then what happens, collision is a indicator that two tags chose the same the value to 0, right to communicate both of them got to 0 and therefore, they had a collision that is also possible just giving you another possibility. Then it goes into nice understanding of the Q algorithm and how the Q algorithm works and all that.

So, you can read this document it also does the mathematical analysis of that system and explains several things in a nice manner.

(Refer Slide Time: 08:28)



EPC Gen2 protocol which is necessary in anti-collision operations. This includes commands that Select a subset of the tag population, and Query the tag population to respond RN16, and ACK to acknowledge the successful reception of RN16 and expect further reception of EPC, and QueryRep to start another slot. Reader model also features inventory sequences and link timing, as shown in Fig.5 and Table I.

This is essentially what I wanted to convey and look at this picture as well, this is a single tag reply, what actually happens you are the interrogator, which is the reader we discussed this earlier, but this is just capturing it all over again. The interrogator gives a select followed by continuous wave then a Query command basically; It is starts with a continuous wave, then a select command selecting a subset of the population of tags then gives a continuous wave and then a Query command with all the related information Q information and so on.

Then RN-16 can be given back by the tag provided a continuous wave is supplied. So, you can see that the continuous wave is supplied here, and RN-16 comes here typically RN-16 there is a delay between the end of the Query command and the start of the issue of the RN-16 itself will be of some length. So, the CW should be sufficiently long to take care of the fact that RN-16 can be communicated back once it receives the RN-16 the reader now issues a ACK back and once the ACK is issued, the tag now gives away its electronic product code and so on.

And then it gives that when this continuous wave is supplied, now Query Rep or Query Adjust if EPC is valid. It is issued and the reader can actually also issue a NAK, if EPC is invalid all this happened, but then it issued a NAK because the EPC was invalid. The electronic product code is invalid. Now, this is a case of collision, you can see that Query and then there is a collision and then there is no reply coming back.

And then there is an invalid ACK and so on. So, I would urge you to read this document thoroughly to understand and get a feel of how exactly communication happens. And this is available for easy download, please do look up this and get a feel for the problem that we have on hand. So that is about this document. But I also have to tell you the other advanced aspect of multiple readers for that I will direct you to one other website. And that is this website, I could not think of a better website which explained several search modes and sessions.

Then what I found here, you must read this, the website link is up here, you can see this, this is the website link, do connect to the website link and read up this document, I will give you an overview of this document so that you get a feel for the problem. Now, this document is actually talking about what you can configure from the for the tags. By the way, this document is actually a support of the standard.

So, if you read the standard, you will actually come to know that there are several sessions that are supported by EPC Gen 2, there is session 0, session 1, session 2 and session 3, there are 4 sessions. And there are two ways by which the reading actually happens. There are two states which are called A and B. You can see that the two states are actually corresponding to written here.

Now the good thing about the requirement for these sessions is as it is already mentioned here, there are two reasons. One is you want to determine when a tag will respond to a Query from the reader. And allows a minimum independent state when communicating with multiple readers at the same time. So, it serves two purposes, multiple readers is an important thing. Otherwise, sessions cannot be if you do not have sessions, you cannot support multiple readers, that is a problem.

And there is in EPC Gen two compliant tag, there are two states A and B for session zero, there are two states A and B for session one and so on, there are states A and B for session three and so on. So, these are states maintained for each session independently and there is a flag, there are in flags for these sessions. Now, any time a tag has to be read, it is usually read in A. And when the tag is in state A it can be in session 0.

And it has read in A, if you want it to be read an B also you must ask moment it actually it goes like this, when a tag is read, it goes from state A to state B. And then it transits back to state A and then it goes back to B, every time it is inventoried it goes to B. But if you do not want the state of being inventoried to be long, then you can ask the tag to quickly come back to A so that it can be inventoried again.

That means you are spending 0 time in B and more time in A it is transiting to B but you are not waiting there at all. You are asking the tag to come back in such a way you actually call it dual target, you are reading it in dual target session zero dual target mode. The problem with dual target mode is you can think of the following this way that you read in A you go to B transit back to A very quickly.

That means the same tag can be ready for being inventoried multiple times. Now look at the problem. If you have a population of tags and you have a reader and you have a population of tags here, some tags are closer, some tags are farther away. And for your good fortune, the signal which is caught by the nearby tags have a better signal to noise ratio and therefore they are able to power up much more quickly as compared to somebody that is far off.

Then this tag has the ability to and if it is said to dual target mode, then you can think of tags which are inventoried, can continuously get keep getting inventoried. So, that problem will happen in dual target mode, you are never giving a chance for the tags behind to be read at all because it is just that they are sinned for being far off from the reader, which is an unfair way, totally unfair method of getting tags that are closer, which have better signal strength.

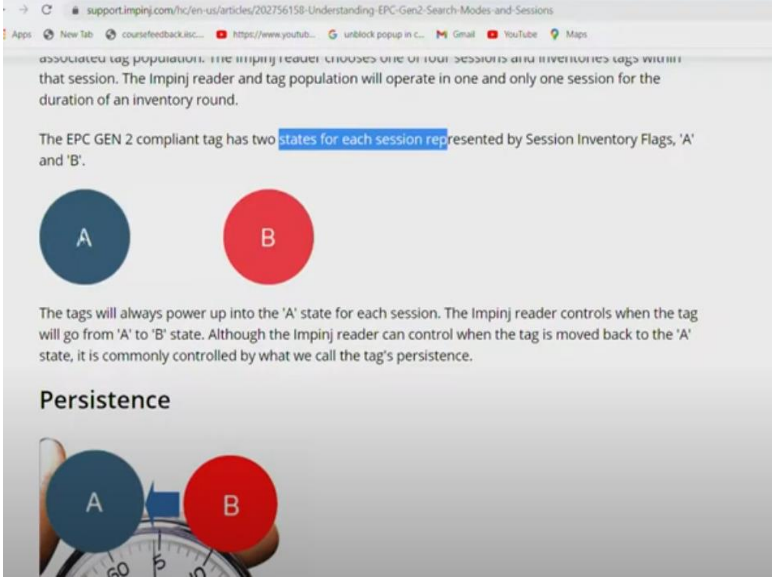
Which get powered faster, quicker and they are able to respond back, keep getting the right to get inventoried many many times. So, these unfair ways have to be removed. And to do that, what you should do is you should, I mean it depends on the scenario, many times you want to also think about why you want to put it into a dual target and why you want to do it in a single target mode.

Typically, if the population of tags are small, you can put it to dual target mode, how small well that is another subjective matter, you have to look at the performance of the reader.

So, that being the case, otherwise, what you should do? You should set it to single target mode in single target what actually happens is, this A state that you see here.

(Video Ends: 16:17)

(Refer Slide Time: 16:18)



support.impinj.com/hc/en-us/articles/202756158-Understanding-EPC-Gen2-Search-Modes-and-Sessions
Apps
New Tab
coursefeedback.iisc...
https://www.youtub...
unblock popup in c...
Gmail
YouTube
Maps

associated tag population. The Impinj reader chooses one of four sessions and inventories tags within that session. The Impinj reader and tag population will operate in one and only one session for the duration of an inventory round.

The EPC GEN 2 compliant tag has two states for each session represented by Session Inventory Flags, 'A' and 'B'.

The tags will always power up into the 'A' state for each session. The Impinj reader controls when the tag will go from 'A' to 'B' state. Although the Impinj reader can control when the tag is moved back to the 'A' state, it is commonly controlled by what we call the tag's persistence.

Persistence

It reads this gives, it inventories itself to the reader and then goes to B state and remains in B state. It remains in B state, when it remains in B state, how long does it remain is another problem. In some cases, it can come back after a few hundreds of milliseconds, in some cases, maybe a few 1000s of milliseconds later. So that, how long it comes back? After being in B state is another thing, we will come to that.

But the good thing is if you are inventoried in A the tag moves for that particular session to B, it now keeps quiet for at least some amount of time. Because it says I have got inventoried let me give chance to others who are far off and get inventoried. So, that possibility exists if you put it to single target mode, but the downside of that is you get only one tag read, I mean the number of tag reads from a given tag is limited because of single target mode.

So, the configuration that you have to do will have to take care of this particular aspect. Now, I will introduce another term which is called persistence, we already discussed that once it is read in target A, in state A, single target mode, if it is single target mode it is a goes to state B and remains there. Now question is how long right that brings us to this question of persistence time. Persistence time depends on the session of interest, this chart will tell you this exactly.

S0 if you set it to S0 you have some time, then you have S1 you have some other time, S2 some other time and S3 some other time. Look at the required persistence chart here clearly telling you how long the persistence applies to different states, I urge you to read this as thoroughly as possible. This is a summary from the main the specification document of EPC Gen two. So, the persistence time is a very important thing.

(Video Starts: 18:40)

Now, dual target I already mentioned these are actually names called single target is just reading in A and putting it to state B and then dual target is reading asking B to come back quickly to A. So that you are getting more tag reads. You see now what he talks about. So, these targets also you have to read in detail. So, each of the modes contain the word target in the name. Target refers to whether the reader will single it that is select only tags that are to be read in a single state or whether it has to be reading A or B.

Single you want or in both states you want to read or if it will singulate tags in both A and B that is called dual target. So essentially, I look at this picture in dual target reads A tags that is the tags to begin with are in state A, one at a time and moves them to B state that is you recall I mentioned about the reading of tags one by one. That is how tags are read sequentially, one after the other one after the other how they are inventoried. We do that picture.

So go back and look at that picture. This is essentially connecting to that, reads A tags one at a time and moves them to B state, then reads B tags one at a time and moves them back to A state, then repeats the above activities over and over again. So, what it actually happens is you are taking from A to B, and then you are taking tags which are in the B state, getting them back to A state and reading them again.

So, reads B tags one at a time and moves them to A state. So, you can see this cycle goes on and on and on, this is called dual target. He mentioned this very clearly. In single target, you do not see this arrow going back arrow, you only see a single arrow that means you read in state A, then you go to a single target. So, you go to B and you don't read here that means you are only reading in one state and therefore it is called single target.

This mode is good for high population dynamic environments, where you want inventory that tag a single time when it enters the read field of view. It provides the deepest possible scan when there is a large population in a static environment. So, you can see in dual target also there is nice explanation you will get generates many reads good for small populations, static environments and so on.

Typically, smart shelf he mentioned also some examples, I strongly urge you to read this document even in further detail, but before I close, I will actually show you this picture and this picture will help you to connect several things. Now, here is the tag in the field which is energized that means reader is throwing energy. The CW has been given a Query select has been given, then CW is thrown Query is given.

Now tags are ready, they can start responding back. Now there are three colors here you can see the left side colour, it is inventory, state A and state B. Now here, this picture holds a lot of information, it says if you do dual target, there is absolutely no difference between the different sessions. There is no way by which you can actually use these sessions in any way if you say if you do dual target.

So, the trick is, dual target should be used in a very careful, controlled manner. And sessions do not make any meaning. If you do dwell target, which is right. You are going from A to B or to forcing it back to come to A. If you, do it like that, then you are actually you are not bringing any great benefit to multiple reader case. So, you chose this here, you see it is inventoried it goes to B, when it is in B state it is read again, then again it is read, it is read again and continuously there are tag reads how many tag reads till the tag is powered;

When the reader removes the power that is it is deenergized not thinking happens here. That is the key point. So dual target session X he calls that means there is no difference between the different sessions if you use dual target, single target is fantastic. See what actually happens in session zero single target you get one tag read and then it goes to state B, it not only remains in that state continuously, almost infinitely as long as there is power, it remains in that state.

Once you shut down power from the reader, the tag loses its power then it goes back to state A this is the beauty. Similarly, in single target session one, you will see that the tag reads are sparse. But it is not as bad as session zero you get one tag read then there is a certain period in which it remains in state B and then it comes back gives you another tag read, then goes back and remains in state B and so on.

Also, when you remove the power it remains in state B it honors the duration of state B this is the beauty really it honors, It could have actually done right at this arrow is not it did not, this little red has extended beyond what it simply means is this is my state B if you remove power in the middle also I will honor the state B. And then I will shift to state A, see the fantastic thing that I can do? This is a very nice thing about session one.

Similarly, in session 2 and 3, there is hardly any difference between the two, except these lengths as you can see, you get one tag read, then you remove power it has so much of persistence, the power that after you remove also, the tag continues to maintain that whatever micro power that it has deterred. So that it can honor this state B, and then only it comes to back to a point where it can be inventoried, whereas singles target with suppression has you know, some other setting here.

So, these are the things that you can easily use in a very nice manner. For example, let us say, that smart shelf application. In the smart shelf application, you could essentially, keep let us say, the reader, which is mounted to the rack can be like this, can be set to single target. You will get one tag read at a time, and then all the 1000 items are being read. Meanwhile, there is a another person who comes in and wants to inventory the item, he need not choose session 0.

He should not he should choose session one, and then come in and perhaps inventory the tags and then go away. So, this gives you an option, because this state B is extended for the significant amount of time for other readers to come in and Query the tag before it can actually before it comes back to its original state. So, this again, is mentioned beautifully here. And you can see that the picture is improved here again, when it is de-energized and comes back with power.

What actually happens is mentioned clearly here. And therefore, the point is that the tags are quite clever. And they do understand the issues of multiple readers. And the functioning is now quite successful with respect to multiple readers. So please don’t forget to go through these examples. That is all I have to say about RFID self-reading, understanding from the places where I have pointed out, very important for a complete understanding of RFID discussions. Thank you very much.

(Video Ends: 28:08)