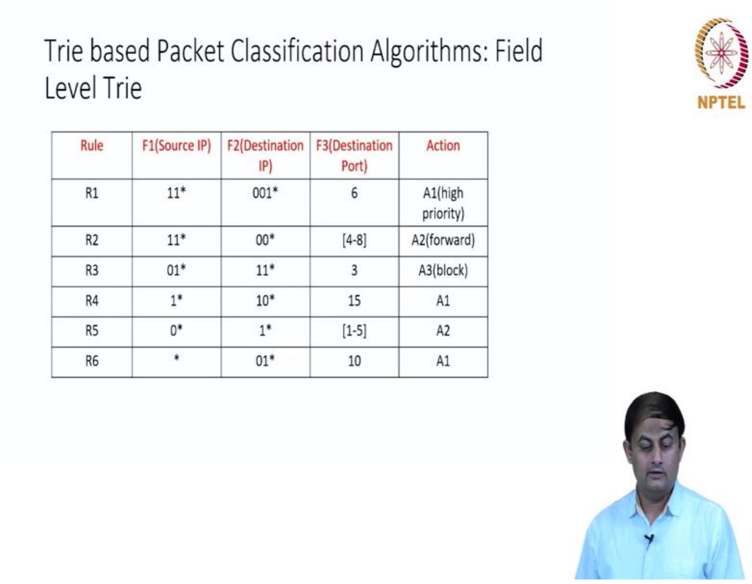
Advanced Computer Networks
Professor Doctor Neminath Hubbali
Department of Computer Science and Engineering
Indian Institute of Technology, Indore
Lecture 13
Packet Classification Implementation - Part 2

We will take an example of that as well.

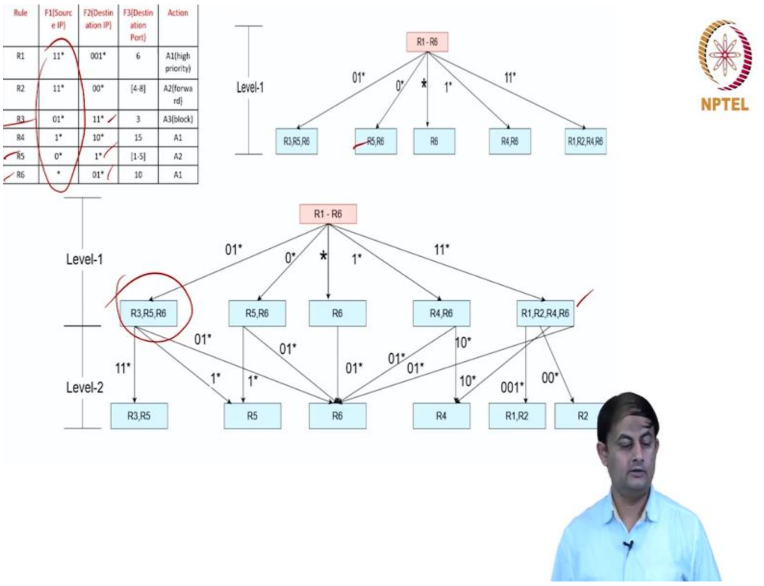
(Refer Slide Time: 00:21)



Rule	F1(Source IP)	F2(Destination IP)	F3(Destination Port)	Action
R1	11*	001*	6	A1(high priority)
R2	11*	00*	[4-8]	A2(forward)
R3	01*	11*	3	A3(block)
R4	1*	10*	15	A1
R5	0*	1*	[1-5]	A2
R6	*	01*	10	A1

So, let us see, we will take a classifier with the six rules. Here is the classifier with the six rules and there are three dimension here, F1, F2, F3. So, F1, F2, F3 is just the name, we can call it as D1, D2, D3 as well, there is no issue with that. And it has got the port range as well and the (())(0:48) as well. So, let us try to construct the three level type of this, the set of T rules for this classifier.

(Refer Slide Time: 00:56)



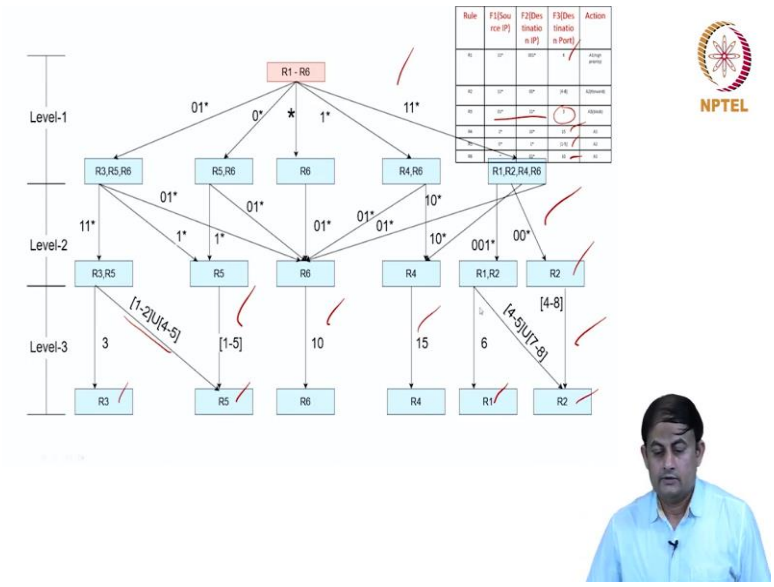
So, as usual, I try to start with the first dimension, so now, every node at first dwell node R1 to R6 is applicable, that is my root node. And at the level 1, what are the unique combinations? I have got these many, 1 1 star, 0 1 star, 1 star and then the star itself and then the 0 star. And that is how they look like. When the 01 star is the prefix rule R3 is applicable, this case and R5 is applicable because 01 star is a subset of the 0 star. So, still R5 is a potential candidate. And both of these are the subsets of the R6. So, rule R6 is still the candidate.

On the input 0 star, only R5 and R6 are the candidates, because R6 is actually superset of the R5. And on the star anyway only rule R6 is applicable. On one star, rule R1, R4 and R6 are the potential candidates. And on 1 1, you got R1, R2, and then R4 and R6 as well. So, R4 is a superset of both R1 and R2, one star and with R1 and R2 are 1 1 star. So, that is how the first level nodes actually look like. Now, again, I take one node at a time, and then apply the possible second level prefixes and then see how the branching look like.

So, here is the second level branching from the node R3 to R5 to R6. Per the rule R3, 1 1 star is one-unit prefix. And then per the rule R5 1 star is another prefix. And per the rule R6 01 star is non prefix. So, that is why, I got two-three branches originating from this particular node in the tree structure. So, I continue doing that from the node, R5 and R6 as well again, there are two branches; for R6, there is only one branch and again from the R4 and R6, there are two different branches, one to the R6 and one to the R4.

And there are three different branches from the, in fact four different branches from this particular node, the second level set of the nodes. So, this is how you construct the type, now all the unique combination for the level 2 are also taken care. Now, let us add the third level dimension into the same price tag and then see how they look like.

(Refer Slide Time: 03:44)



So, from the second level division R3 to R5, what it says is when the rule R3 is applicable the port number need to be 3. So, only then the rule R3 is applicable. So, the level three I added a node level R3 and edge level 3 and for the remaining cases, so what is the rule? R5 says it is 1 to 5 but 3 is more specific than the 1 to 5 range and R3 is prefix, so these two are also super sets of the R5 as well. So, three is more specific. So, rule R3 is more specific, I separate it out; for the remaining set of the cases, what are those?

When the port number is 1 and 2. And when the port number is 4 and 5, I need to use this rule R5 to classify that particular packet. So, that is why I wrote this 1 to 2 union go to 5 as one branch for this particular node. And I continue doing this. So, for the rule R5 again 1 to 5 is there, so that is fine, for the rule R6, only 10 is required, so that is fine rule, rule R4, only 15 is required, that is fine. And for the rule R1 and R2, there is an overlap. So, R1 is a subset of the rule R2 when the 6 is there. So, I am going to take the, I am going to classify that particular packet with the rule R1, that is what the label.

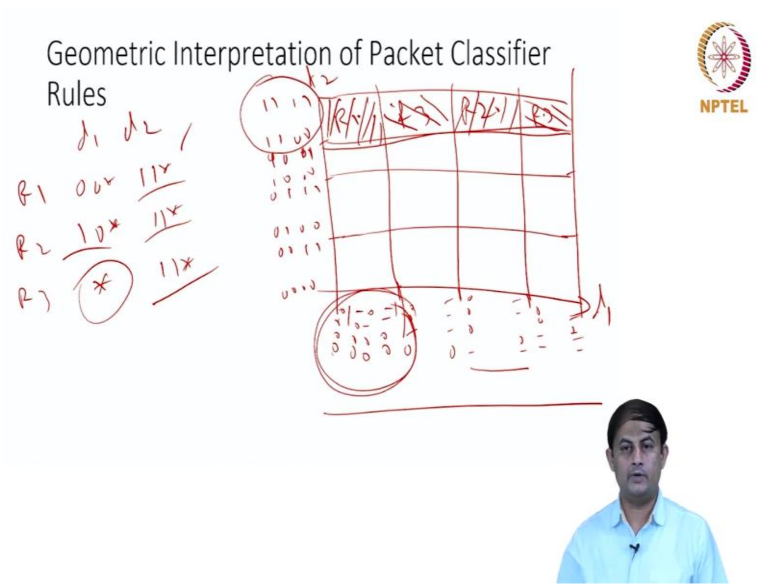
And when the cringe is in between 4 to 5, excluding the 6 and then 7 to 8, I am going to apply the rule R2. And similarly, from this branch, the entire 4 to 8 range when the prefix, the cost level is 11 star and 00 star, and then the in between port number can be anywhere between 4 to 8, then I apply this particular rule or to do the classification. So, that is how you segregate. If there is an overlap, then you look at the more specific rule sets and give them the preference, take out that plan separately and the whatever is the remaining portion is there, that is actually used for the other possible rules, that are there in the perspective.

So, that is how we actually construct the field level try. This is basically, you start with all possibilities, then slowly segregate what are the rules that are potential candidates applicable for this particular prefix combination and the port number combination in the search. So, with that, we will end the discussion on the tree structure, and we move on to the another kind of the interpretation for the same packet classifier.

So, this interpretation, this is based on the geometric notation of the classifier, what we did is? Assuming all the dimensions are having the prefix values, prefix values meet up the dimension, so, you can think of this as multi-dimensional hypercube and then if a packet comes and the classification axon is basically finding out which of the hypercube in the entire structure, multi-dimensional structure, this point is actually particularly falling in.

So, that is what, there are nice geometric algorithms, which can actually find out in which of the hypercube, the particular point is actually falling into. So, although we do not look into those geometric algorithms as such, but what is the notion behind how do we interpret this classifier rules into that geometric interpretation that we will take look in this particular discussion.

(Refer Slide Time: 08:10)



So, let us say again, we will take a simple example, on the two-dimension D1 and D2, and the classifier has got rule R1 has got 00 star, and then a 11 star and rule R2 has got 10 star, and then 11 star, and rule R3, let us say this is star and then 11 star. So, what I do is? There are two dimensions and now I need to look into, this is a point falling in the two-dimensional space. So, let me draw one dimension for a particular this, let me call this as D1, and then this dimension D2 is on the y axis.

So, I am going to divide this x and y axis into several ranges and looking at the prefixes I am going to mark this particular area is actually falling into this particular prefix. That is what the geometric interpretation is. So, here, I will start with, on the x axis, a value, 00 00 and then divide this into four phases. From this area to this area, the values with prefix 00 and these two bits are varying, so I have 0001 0010 in between and 0011. So, the moment I crossed this area, I got 01 and then 00.

This will continue up to 01 and 11. When I call this area, this is 100 00, this is 1011, 1100, 1111, R1, this is the division for, the first dimension is D, prefixes actually fall into one of these regions. So, for example 00, where exactly the region 00 is falling. So, particularly in this range. These four values are having the prefix 00 star. So, these four values. Similarly, on the y axis as well, I am going to create such similar division, this is 00 up to 11 0100 and 0111 and this is 1000, this is 1011, then 1100, 1111.

And I can draw a line from here to here, here to here, here to here and here to here and also the vertically from here to here. So, this is a two-dimensional grid. And what it is saying is looking at the prefixes on the dimension D1 and D2, we need to identify what is the particular region on which this particular point is actually falling. So, for example, let us take the case of R1 00 star, and then the dimension D2 is 11 star. So, where exactly on the dimension D1 00 star is appearing?

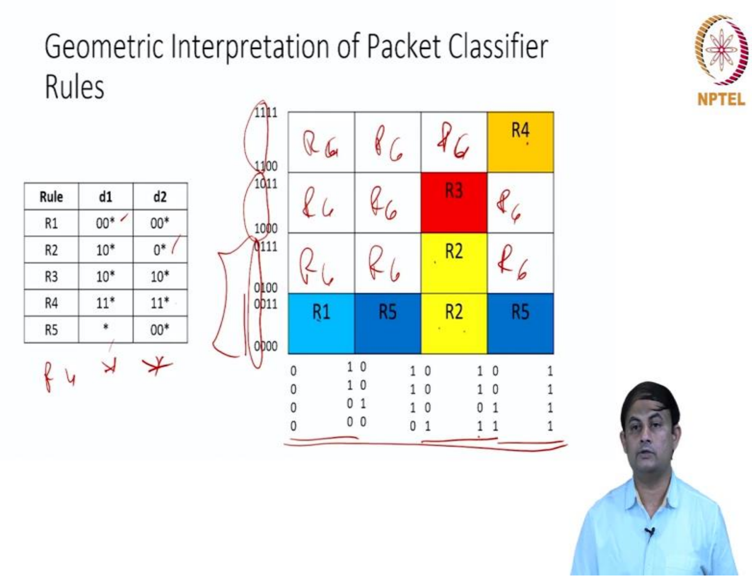
So, it is these four values are or this particular range from 000 to 0011, so, this range is actually indicating 00 star and the second dimension on the y axis, it is saying one 11 star, so, where exactly one 11 star is? These four values are indicating 11 star. So, I am going to mark this region as the region for the rule R1. See, if the point falls in this particular box, or the two-dimensional space, then I am going to classify that with the rule R1. So, this is the place where the R1’s rule is applicable for that particular packet.

Similarly, for the rule R2 says the dimensional D1 says 10 star. Where exactly is 10 star? This is the region where 10 star is there in the x axis on the dimension D1 and the second dimension is still 11 star, 11 star is the same region. So, this is your rule, R2, this entire range is R2. And if you come to the rule R3, Stephen says this is Star. So, what is star? Everything from here to here is star and but the dimension D2 says still it is 11 star, 11 star is the entire this region is actually 11 star, but two of them, R1 and R2, have already occupied two spaces, they are more specific than the rule R3, the remaining part two boxes, I am going to classify R3 with the...

So, if at any point of time, the more specific rules are labeled, so if there is an overlap, here in this case, R3's range is from here to here. But R2 happens to be a subset of that entire range. Whenever the point falls in this particular range, I am going to do the classification with the rule R2. Whenever a point falls in this range, I am going to do the classification with the rule R1. But when the point falls in this in this box, then I am going to use the rule R5, that is what the geometrical interpretation is.

So, you can extend this to three-dimension, four dimensions. So, although drawing on this page with the four dimension is not possible to indicate but you can imagine, just like the machine learning people visualize this as the hyper dimension. So, two dimension, three dimensions, n dimensions, you can go and then in which of the hyper plane the point is actually falling based on the values in the packet, then you decide to which rule to apply. Once I label the hypercubes with the appropriate name, then rule set, then I can actually use that rule to do the classification.

(Refer Slide Time: 14:33)



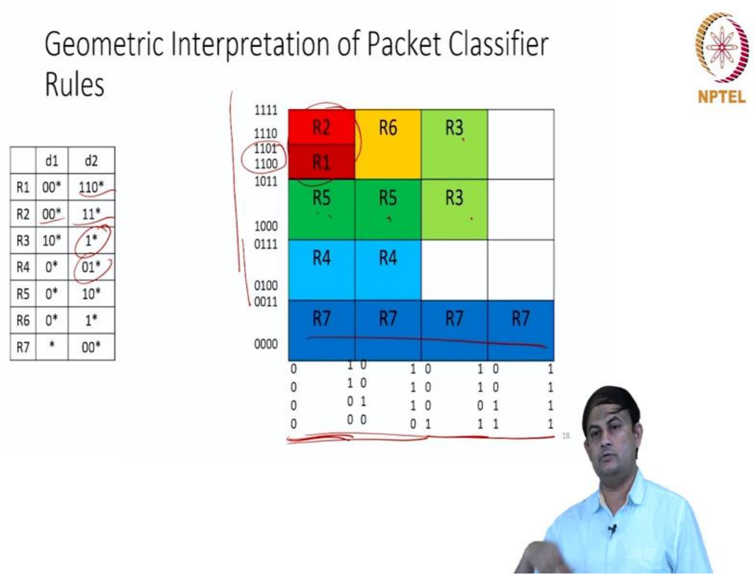
So, let us take one more example. Here is the classifier with the five rules, from R1 to R5 and these are the prefix values for the dimension D1 and D2 and when I do the classification on the two-dimensional grid, this is what the labeling look like. So, rule R1 is 00 star, so this is the 00-star region and the dimension D2 is also 00 star, this is the region where it is 00 star. So, that is why I labeled it as R1. And R2 says 10-star, 10 star is here, and what the dimension D2 just the 0 star.

So, in all the four, these two boxes particularly because this is changing, only one bit is constant across these 4 colors, which is zero. So, I take this as the region, so I label this and this box as the label are R2. Similarly, R3 is having 10 star. So, the same 10 star and the second level is 10 star. So, 10 star on the y axis is this region, that is why you need to label this as R3. And when I go to the R4, this is 11-star, 11 star is in this region and on the dimension D1 and similarly on the dimension D2 also it has 11 star, 11 star in the dimension D2 is this one, that is why this is labelled as R4.

And when it comes to our R5, this is Star. Star indicate the entire range, on the x axis, but the y axis is 00 Star. 00 Star is these four values itself. But R1 and R2 already occupied the space, they are more specific than the R5. So, the remaining two boxes, actually I label with the R5. So, that is the labeling. So, it is not necessary that here in this case, some of the boxes are not labeled with any of the rule. So, if a packet comes to those attributes, I will not be able to do the classification.

So, this classification is not complete, meaning for all combinations of the values that you can find, you do not get the labeling done. So, normally, in order to make this complete, what we will do is, add a rule which is more generic, something like this, that we call it as the R6 and the star star, anything does not matter, whatever is there, so you just label. Once I do the labeling for the rule number R1 to R5, they are already there, remaining all boxes are now labeled with the rule R6. So, that is how the classifier are made complete. So, if we have such a rule, if every box is labeled, then the classifier is said to be complete. So, I can write R6 in the remaining all boxes. Now, this classifier is complete.

(Refer Slide Time: 17:44)



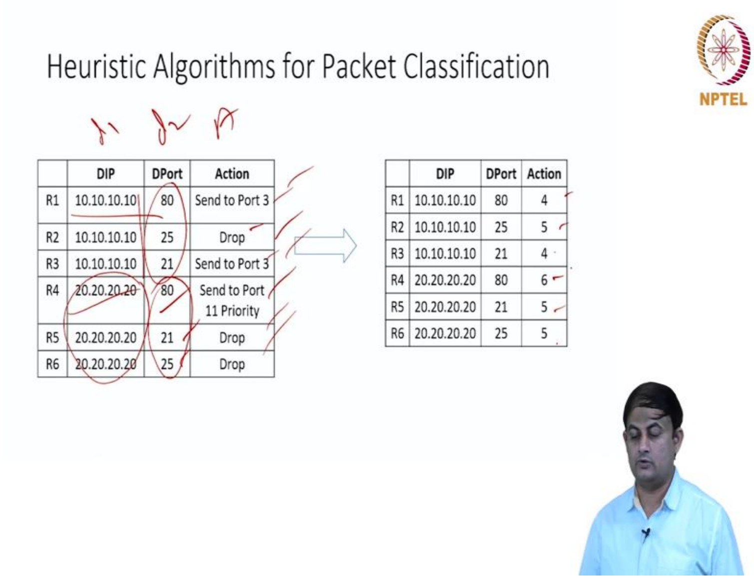
So, maybe we will take one more example. Here is a classifier with the seven rules, I think we use this example earlier as well and if you take this plausible seven rules how the on the two-dimensional space the labeling would look like, again, rule R1 is 00 star and on the dimension D1, that is this space and the dimension D2 is 110 star.

So, you can see this 110 is only for these two cases values the 110 star is applicable, so that is why your label this will go with only here and but for the remaining cases R2 is applicable, it is still R2 is on the dimension D1 is 00 star which is still this space, but the it is 11-star, 11 star is applicable in the entire range.

But 110 star is a subset of 11 star. So, that is why, this is more specific than I have labelled it with R1. The remaining space of the same box is labeled with the R2. And similarly for the R3, it is 10 star, 10 star is here and on the y axis, this is 1 star, 1 star is applicable from here to here, that is why these two boxes are labeled with the R3 and for R4, this is 0 star, 0 star is applicable for these two cases and then 10 star sorry, R4 this is 01 star, so 01 star is applicable in this range, these two are labeled with 01 star R4 and for R5, it is 10 star and with 00 star, so these two are labeled with the R5, and for R7, this is the star, entire range and then the y axis is 00 star.

So, these 4 boxes are labeled with the classifier rule R7. So, that is how you will do geometric interpretation for the classifier. So, that is one kind of thing interpretation that you can print and if you have an algorithm which can find out in which of the equals Q space, the box, the point is actually falling, point means the attributes, whatever you are picking from the packet and where it is actually falling, you actually use that particular rule whatever action that particular rule says, you apply that rule to that particular packet. So, you can use this interpretation to do the classification.

(Refer Slide Time: 20:30)



And we also saw some of the heuristic mechanisms put earlier as well. But those are not the only heuristics that you can use. There are other interesting heuristic-based algorithms that we can use to do the packet classification mechanisms. So, here is one, let us take the simple example. Here is the classifier with the six rules and there are two dimensions.

So, dimension D1 is the destination IP and dimension D2 is the destination port number, and this is the action that you need to take on this particular rule set. So, what optimism that this heuristic algorithm brings is, it is based on the observation that some of the field values for certain rules are common. So, for example, rule R1 to R3 has the same set of the destination IP address, but with change of the port number values.

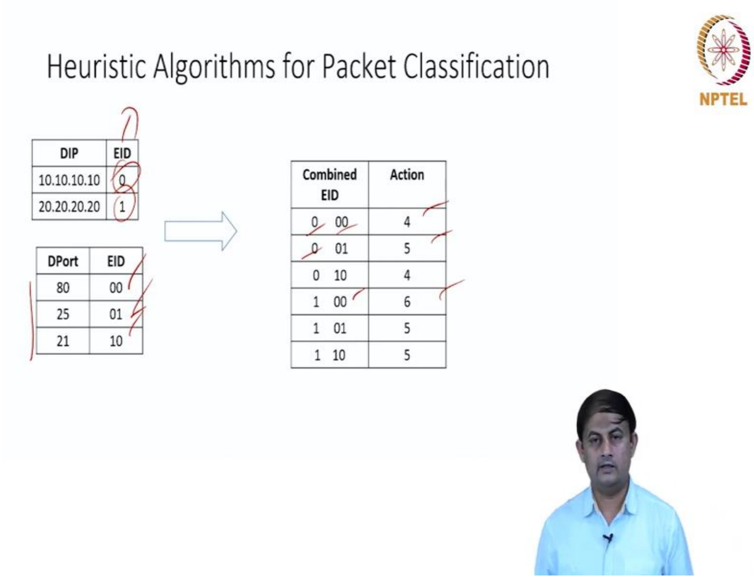
And similarly rule R4 to R6 has got the same set of the destination IP address, but the port numbers are changing. Can we exploit this kind of mechanism? This is not very uncommon. If you have for different applications, you want to treat them differently for the different port numbers want to assign different priority and action, as specified in this accent list, when the port number on the same destination IP address when it is at 80, then you takes on this action, when it is the port number 21, you drop, you put on 25, you drop, so rules can actually specify such kind of the differentiation.

Now, can I exploit this kind of rule sets in my classifier and build some kind of the structure which actually allows me to do the lookup quite quickly, number of the comparisons quite quickly and then do the classification or to decide which action to take. So, given the classifier is there, and the rules are there, what I do is, I am going to extract the unique destination IP pins or the prefixes from that toolset and then if I, in this case, if I want to do the matching when the IP address 10 dot 10 dot 10 dot 10 is there, this is a 32 bit field, but it is repeating three times, can I use this comparison to instead of doing if this one fails, then I need to go and examine this one and this one and so forth.

So, can I reduce the storage structure? This is not going to not only take the storage structure more, I need to store this structure three times because the IP addresses appearing three times and then the port numbers as well. So, I can bring a slightly optimized version of this question which is by instead of storing the IP address three times, I am going to store one time and then build, take an index to the second level table and using the label in the second level table, I am going to do the classification.

So, let us take a look at how it looks like. So, I am going to transform this every action that I am having with this particular action field, I am going to assign a number. So, what is that number sent to port number 3 is the action number 4, drop is action number 5. So, wherever drop is there, I am going to write 5 and then send to port number 3 is action 4, and then send to port number level with the priority is action number 6, so their labeling is done.

(Refer Slide Time: 24:35)



Once this is done, the next level step that I am going to do is take the unique IP addresses in the first level field and then create a table for every unique destination IP address, an equivalent ID is assigned. So, for example for 10 dot 10 dot 10 dot 10, I am going to assign the ID 0, for 20 dot 20 dot 20, I am going to assign an ID of 1. So, now, the table structure, earlier I had to compare this with the 6 number of T entries, if I go linearly one by one.

But now what I did is by writing this, there are only two entries, because many of them are many times the same entries are repeating. So, I am going to assume whenever the equivalent ID is 0 is going to be 10 dot 10 intermedius 10 dot 10 dot 10 and then I take the further action. So, this is for one dimension and for the second destination port number also I will do a similar structure. So, destination port number unique destination port numbers are these three, the IP address are 2 but the port numbers of 3.

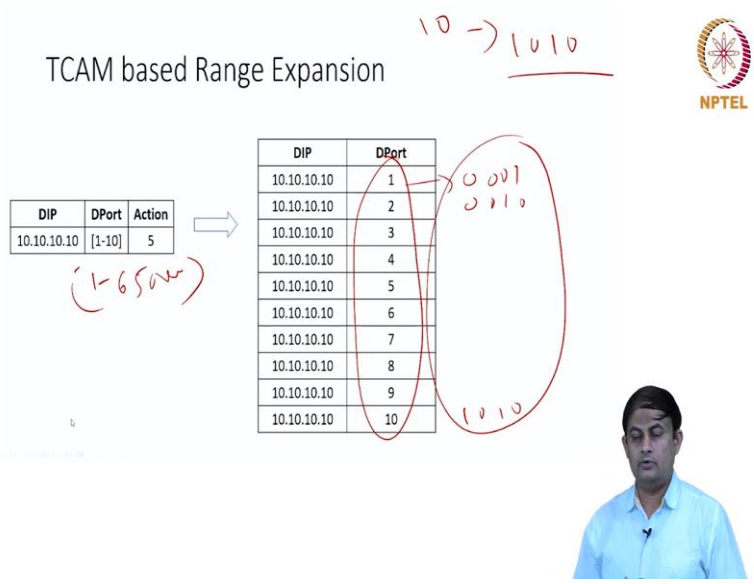
So, every time I look at the nearest two digit, n digit binary number which is required to assign, uniquely assign a number, an equivalent ID for this port range, so, for 80, I am going to maybe assign 00; for 25, I am going to assign 01 and for 21, I am going to assign 10. And now I will combine these two and we will determine which is something like this. When the first A will 0 and the second dimension label 00, 00, I take the action 4; when the first level the equivalent ID 0 and the second level, this one is 01, I am going to take the action number 5, these are coming from the previous table that is there.

So, we just assigned these 45465 number and using the same number I am now combining and labeling. So, when the first level label is one and the second level label is 00, you take the action 6 and so forth, I am going to give the construction. So, by doing this kind of the alteration into the table, what I am bringing is actually the table which is occupying more number, the more the number of the repetitions you have in any one of the dimension, the more larger the table structure look like.

So, instead of that, if I assign the equivalent ID, I am going to compress the table and occupy less space on the router and you can store a larger classifier structure in the table, that is one heuristic that you can bring. And if you do a first level application as long as unique combinations are limited, there will be a substantial reduction both in the first level table and also in the second level table.

So, second level table in the sense this combined table since you are going to assign the binary numbers, unique numbers to those fields, which are of 32 bits and 16 bits and you combine them into only using only maybe 2 or 3 bits or whatever the number of bits required, then the second level table structure will also substantially reduce and it is more quicker to search in this second level table, thereby you bring the speed up the operation. So, that is one another heuristic based mechanism you can use to do the backend classification quite quickly.

(Refer Slide Time: 28:35)



And the last thing that I want to discuss is again, if I want to do the lookup or the classification on the hardware, we understood that earlier T can be memory, the one which is used and it can work very well with the, when the dimensions are the values are in the form of the prefixes. But interestingly what we noticed is the values inside some of the dimensions need not necessarily be in the form of the prefix but they can take the ranges particularly the port numbers.

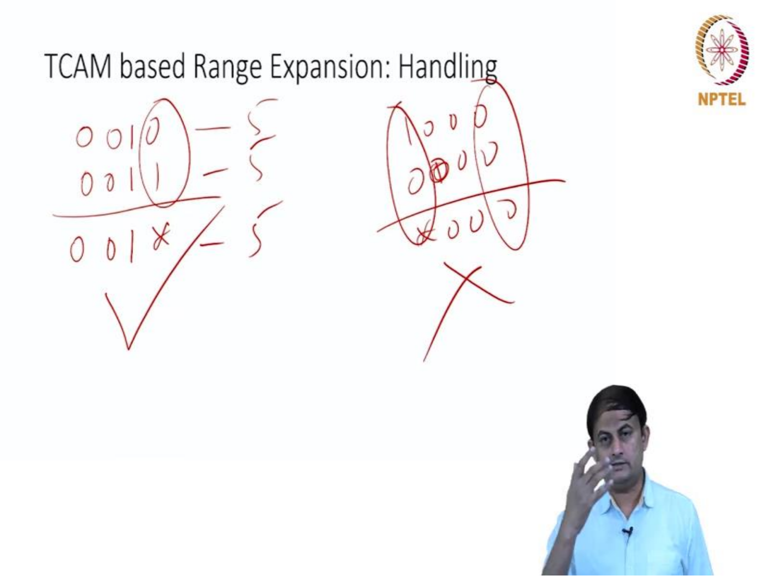
So, this leads to a very challenging problem called as the range expansion. So, if I want to convert the ranges into prefix format, if I have some, let us say the port number is 10, it is 10 then I can convert it into a binary number. So, the binary equivalent of this one is 1010. This itself although there is no star here but this I can treat it as the prefix format; as long as it is in the binary format, it is okay. But since the destination port is not a unique number 10 but it is in this range, in order to convert this into a prefix format, what I need to do is to expand this range with individual rules.

So, as many number of the what is the range that is special 1 to 10, those many times the rules are replicated, that is where the enhancement of peak actually takes place. This is called range expansion. So, I am going to replace one rule and have an equivalent rule for every unique destination port number in my classifier, and you basically create 10 rows, and then convert this destination port into a binary followed into prefix format, this I can write it as 01, this I can write it as 0010, and so forth up to 1010.

And instead of writing 1234 10, I am going to write these values in this space. So, that is the destination IP address is also in the prefix format and also the destination port is also in this prefix format, then I can use the TPM memory to do the comparison quite fast. Now, the question is, this range expansion problem can be quite tedious to do when you do the expansion.

So, for example, if the destination port number is between 1 to 65,000, to something like this, then there are 65,000 rules you created inside my classifier. One classifier one rule, actually exponential 65,000, that is actually quite a large number of rules. So, this is going to be challenging, both for the storage perspective and also for the comparison perspective. So, we need to handle this range expansion problem.

(Refer Slide Time: 31:40)

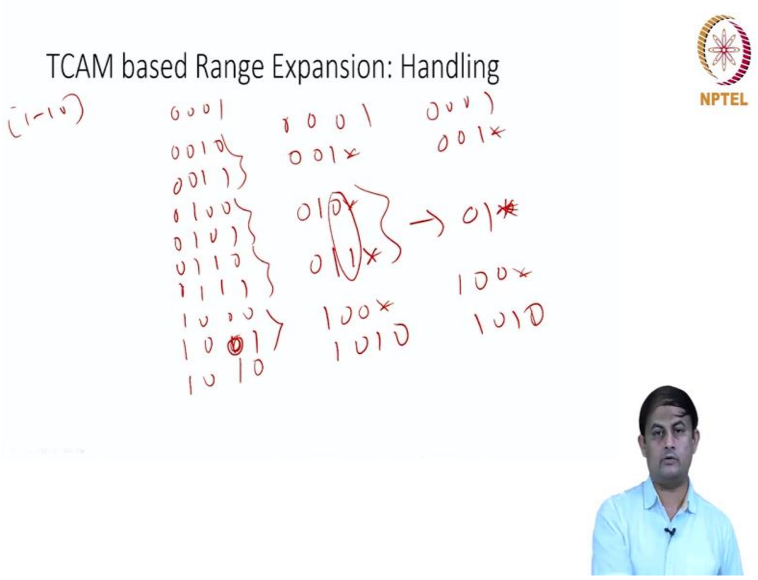


And one way to do the handling that expansion is to compress, to look for the opportunities to merge the rules inside the table and then see if as long as the action taken on that one is common, you can keep merging the rules which are adjacent, and then try to bring down the number of overall number of rules.

So, let us say take the example what kind of the prefixes can be merged. So, here is one prefix, which is 0010. And the second prefix is, let us say 0011. And these two are differing by one bit. As long as the action taken for these two cases is common, what I can do is I can replace these two rules with a star and then we will take the action 5. I am going to basically look for such merging possibilities, as long as the action taken is common, I can keep merging them and get a prefix which is smaller and also the overall number of rules will be reduced by virtue of that.

What I cannot merge is if I have a rule, which is in this format, although these two are differing by one bit, this, I cannot replace it by, so only the least significant bit in this area, which is this is not possible. This is okay. But this is not okay. So, like this, I am going to merge the adjacent rules inside my classifier and then get a reduction on the number of rules.

(Refer Slide Time: 33:18)



So, let us go back to the same example if my port number range is in between 1 to 10 how many number of the unit prefixes that I am going to get? So, as I said, there are 10 rules that are created for this one. So, let me convert those into binary. 001 0010 0011 0100 0101 0110 0111 1000 1001 10. Okay, so, this is the set of binary values that I get. Now, I can see what are the rules that, what are the adjacent values that I can merge. So, for example, these two can be merged, they are differing by one bit, I get a smaller prefix 001 star and these two can be merged to get to prefix 010 star and these two can be merged to get to 011 star and these two can be merged to get 100 star and this is 1010 and this is 0001.

So, from the 10 prefixes and the previous case, now we landed up into to 6 prefixes. Now, are there any further comparisons possible? So, there is one more possibility. These two can be merged to get 01 stock because these two bits are erring; as long as the action is common, I can get this prefix as 01 star and this is 0001 001 star and this is 100 star and 1010. So, these are the 5 unit prefixes that I get as long as the action for the all of them are common I can use these 5 rules instead of the 10 rules in the table, this is called as the TCAM range expansion handling.

(Refer Slide Time: 35:32)

So, how the table look like. So, I started with one rule and destination port number was in this range, I expanded it to 10. And now, by virtue of merging the prefixes in the previous example, as I wrote, so, you get these 5 unique prefixes and I am taking the action 5, that is how you actually reduce the number of the inputs to be stored in the TCAM memory and by virtue of that you are able to accommodate more number of the rules in the classifier and you can still use the TCAM memory to do such operation, which rule to apply, that is exactly the same as that the lookup operation which we have already seen in the previous lecture. So, that is how you handle the different rules and do the classification.

So, let me do a quick summary of what we studied. Classifier is a collection of rules, classifier is denoted with a capital letter C, and every rule has got a predicate and an action. And then the predicates are coming from the attributes of the particular packet, be it the source and destination IP address port or port numbers and so forth. And we said that there are different ways you can build different data structures to do this classification action, one is the tri-based structures, we saw the hierarchical tries and couple of optimizers in particular the set 2 try and the grid try.

And then we also saw it different kinds of try structure which is doing every classification based on the unique attributes in every field and then that is we call it as a field level try and then we also looked at the heuristic approaches to do the further optimization. And finally, we looked at T, how can we use the TCAM based memory and how to handle the range expansion or when we do the, when we convert the rules with port number ranges into the graphics format. So, this is all about the packet classification. So, in the next lecture, we will come back and discuss something else.