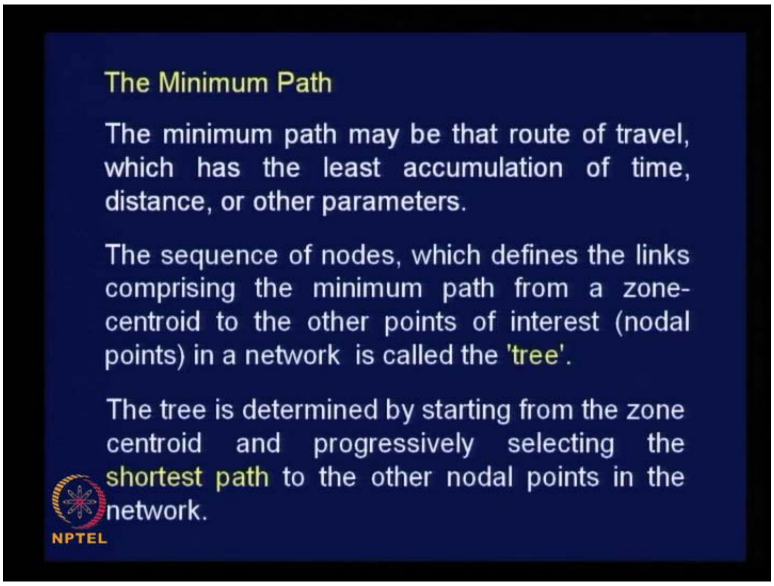
Urban Transportation Planning
Prof. Dr. V. Thamizh Arasan
Department of Civil Engineering
Indian Institute of Technology, Madras

Lecture No. # 29
Route Assignment Contd.

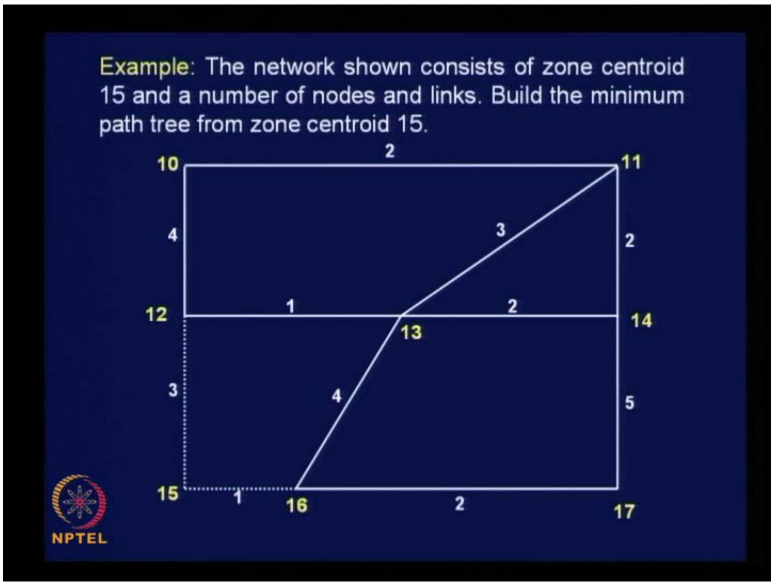
This is lecture 29 on urban transportation planning. We are in the process of discussing the fourth important analytical step in the transportation planning process namely route assignment. You may recall in the previous class, we discussed about route choice behavior by travelers. We learnt about three principles of route assignment based on user equilibrium, system equilibrium, and stochastic equilibrium. And finally, we understood that in most cases, user equilibrium is the basis for traffic assignment in transportation system network. Once we decide on the method of assignment, then we are cleared at the users will follow the minimum travel time path or the minimum path involving the generalized cost of transportation. This indirectly implies that we need to identify the minimum path from each of the zone centroid to all the other point of interest from that particular zone.

(Refer Slide Time: 01:48)



So, that is how we discussed about determination of minimum path from a given traffic zone right, and in that connection only we discussed about the minimum path tree as well as the definition of the minimum path itself. The minimum path may be that route of travel which has a least accumulation of time, distance or other parameters as and only considered. As I said that the other parameters could be travel cost, comfort, convenience, and so on. The sequence of nodes which defines a links comprising the minimum path from a zone centroid to other points of interest, that is very important to the other points of interest. The points to which this particular zone centroid is actually connected in a network is called the tree or the minimum path tree. The tree is determined by starting from the zone centroid, and progressively selecting the shortest path to the other nodal points in the network.

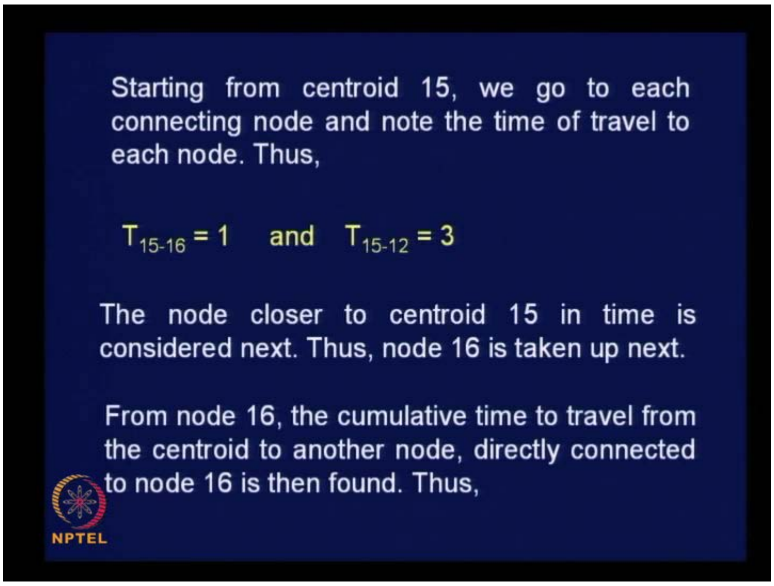
(Refer Slide Time: 03:10)



We start from the zone centroid and proceed along the shortest path and we have considered this example to understand building of minimum path tree. The problem is built the minimum path tree from zone centroid 15 to all the other nodal points shown here. As you could see this is not a complete transport system network, it is just a small segment of a road network link to zone centroid 15. It is just to understand the principle of building minimum path tree. To distinguish between zones and centroid other zones you can see 15 is connected by dotted line and all other links are continuous lines. Let us see how to proceed, further to build the minimum path tree. It is a simple principle of identifying the minimum path.

We start from 15 and there are two connecting nodes to 15, the nodes are 12 and 16. The travel time to 12 in minutes is 3 minutes and travel time to 16 in minutes is just 1 minute.

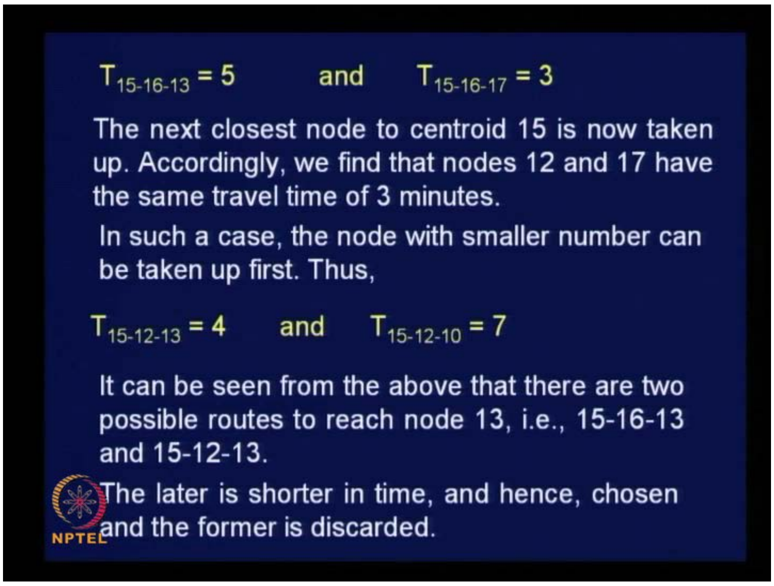
(Refer Slide Time: 04:26)



So, that is how we are going to proceed starting from centroid 15, we go to each connecting node and note the time of travel to each node. Thus, we can write travel time from 15 to 16 indicated here as t 15 dash 16 is equal to 1 and t 15 t 12 is 3 and then the node closer to the centroid 15 in time is considered next. Why considered next? Why should we consider the node closer to the centroid first? Why not the node for that to the centroid first? Any response, it is a shortest distance we have to be calculated sir. Yes, that is the logical reason. Anywhere, we are interested in shortest path why first taking the node which is farther away. But, it does not mean at the beginning itself that, the farther node is not going to fall on the shorter path.

We may not able to say at this stage but, just some algorithm has to be followed to complete the process. It is just conventional to go to the next shortest node, it is not mandatory in fact. From node 16, the cumulative time to travel from the centroid to another node directly connected to node 16 is then found. So, 15 to 16 then proceed further from 16 and please perceive the network clearly.

(Refer Slide Time: 06:30)

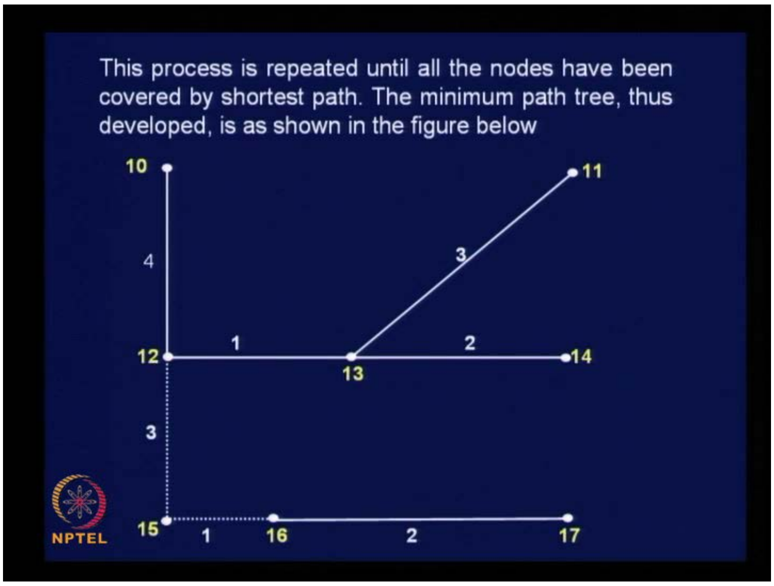


So, if you proceed from 16 is connected to 13 and 17. So, but the path from 15 to 13 through 16 takes about 5 minutes t 15 dash 16 dash 13 is 5 minutes and t 15 dash 16 dash 17 is 3 minutes. The next closest node to centroid 15 is now taken closest in terms of the cumulative travel time. Accordingly we find that nodes 12 and 17 have the same travel time of 3 minutes 12 and 17 and you may have to go back to see the travel time of 12 15 12 is also 3 minutes and here we find 15 16 17 also has 3 minutes as travel time. So, there is a tie, even if you apply the norm of choosing the node, which is the closer 1 2 the starting node we find there are two nodes with equal timings. So, there should be some conventional algorithm to take care of this kind of situation.

In such a case, the node with smaller number can be taken up first it is only a suggestion can be taken up first smaller ending node number. We have in the previous case 12 as an ending node here the ending the node is 17. So, obviously we can proceed from 12 to work with the algorithm. So, we find out the cumulative travel time from 15 to 12 and then to 13, 12 is connected to 13 and it is 4 minutes and t 15 12 10 because, 12 is also connected to 10 and we have 7 minutes as the cumulative travel time and it can be seen from the above where there are two possible routes to reach node 13 they are 15 16 13 and 15 12 13.So we can reach 13 by 2 alternative routes, which route you will choose. 15 16 13 cause you 5 minutes of travel time and 15 12 13 takes only 4 minutes of travel time. So you have to obviously take the shortest path.

So, 15 12 13 is the path to reach 13 from zone centroid not 15 16 13 clear that means, 16 13 is not the link, which is going to form the minimum path tree. The later is shorter in time and hence chosen and the former is discarded later is 15 12 13 and former 15 16 13.

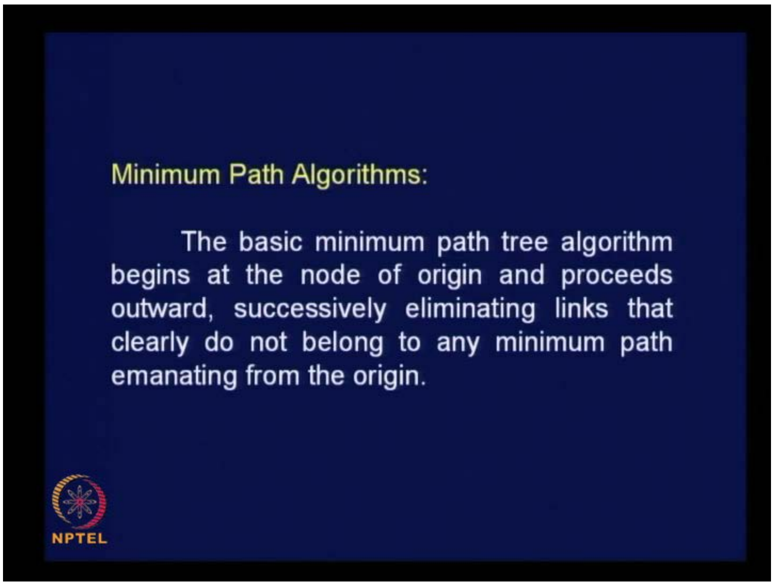
(Refer Slide Time: 10:15)



This process is repeated until all the nodes have been covered by shortest path. The minimum path tree thus developed will be as shown in the figure. This is a minimum path tree. We can see that a number of links have become redundant and hence removed from a minimum path tree, this is what we looking for when you assign traffic you can assign using this network the rest of the links are useless as far as this particular zone centroid is concerned. If you take minimum path tree as the basis for traffic assignment so extend this methodology to cover the whole of the network. And please remember, we are going to work with traffic assignment or route assignment using computer program because the volume of data to be handled is huge and the number of nodes and links involved will be in thousands. In that case, it is not possible for us to draw sketches like this and try to find out the cumulative travel time.

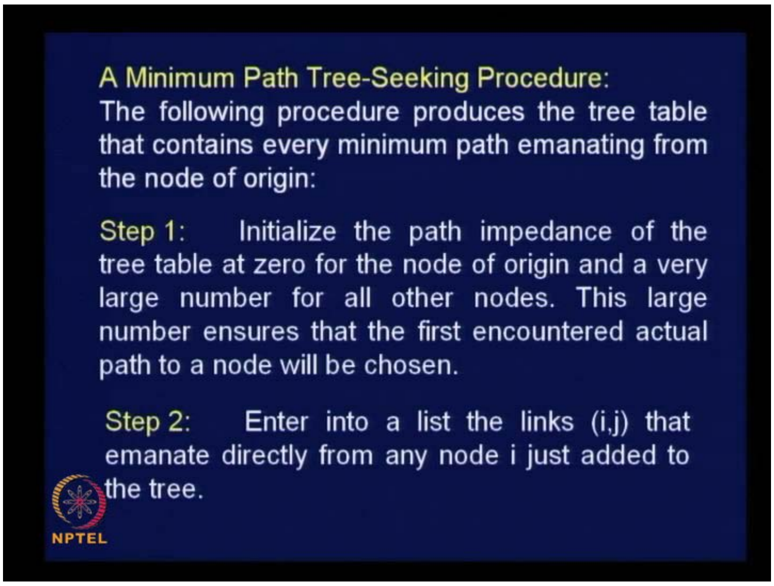
Obviously, we must work with some other methodology what is the possible methodology? Any suggestion. How did we represent the road network for the purpose of analysis? Do you recollect the discussion about link array table, link array? We represented the whole of the road network using a matrix. Link array you can be used to represent your network, use a same link array to identify the minimum path tree also that is how a program can work, program cannot work with figures like this. So we must know how link array tables can be made us off to identify minimum path trees, which are very simple only point. We remembered is the algorithm step by step, we should remember the algorithm then the rest of the calculations will be automatic.

(Refer Slide Time: 13:03)



So, minimum path algorithm, the basic minimum path tree algorithm begins at the node of origin obviously, node of our interest node of origin where, from we start and proceeds outward successively eliminating links that clearly do not belong to any minimum path emanating from the origin. It is a very general statement, if even you are not very clear about the statement you will be able to understand better, when we take up an example just at least have some vague understanding of what is stated here. We are just eliminating links that clearly do not belong to any minimum path emanating from the origin.

(Refer Slide Time: 14:05)

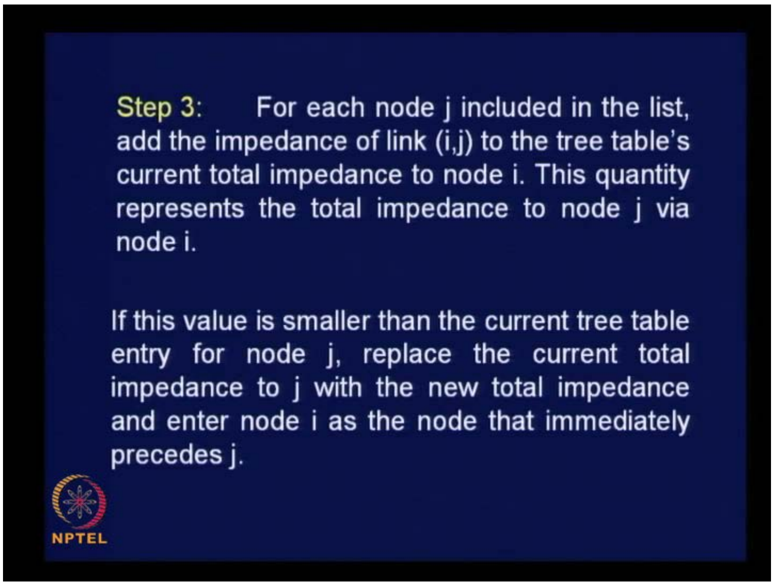


The minimum path tree seeking procedure is as follows the tree table that contains every minimum path emanating from the mode of origin is, what we call as the minimum path tree seeking table and step 1 is as follows. Initialize the path impedance of the tree table at 0 for the node of origin and a very large number for all other nodes. What you mean by this? Initialize path impedance, path impedance is nothing but, in simple terms travel time. So, we are talking about the travel time required to reach that particular node that is, what is mentioned here as path impedance of the tree table at 0 for the node of origin. Obviously, for the node of origin path impedance can be taken as 0 because, we are going to start from there.

And very large number for all other nodes, when we say very large number normally it is taken as to be on the very safe side infinity. Take the travel impedance as simply infinity for all the other nodes and take the travel impedance as 0 for the node of your interest, which is the origin. This large number ensures that the first encountered actual path to a node will be chosen from the origin. We will see how it really works, after taking an example. Step 2, enter into a list a links i j that emanate directly from any node i just added to the tree. Enter into a list the links i j starting node ending node details i and j details that emanate directly from any node, I just added to the tree, just added to the tree. I will just explain to you more clearly or in detailed when we take up the numerical example.

What you understand need to understand here is, just we indicate the numbers of the starting and ending nodes for the concerned links with respect to the considered or the node of our interest.

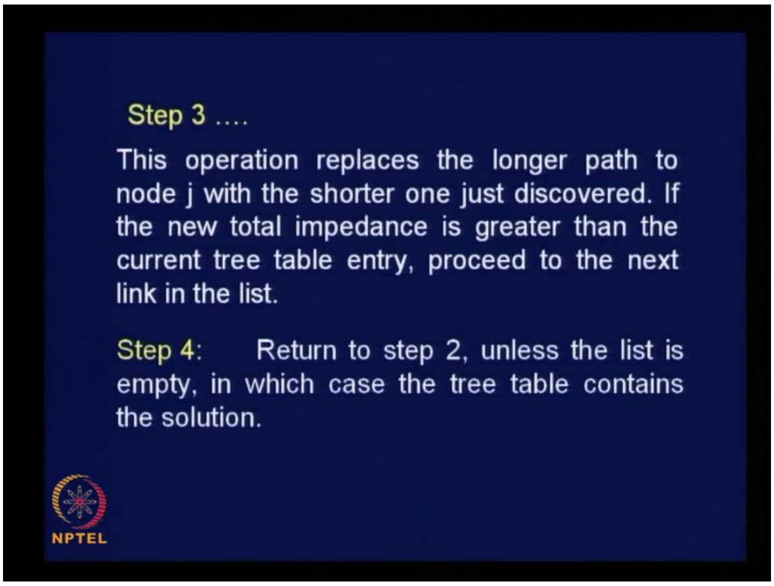
(Refer Slide Time: 17:11)



Step 3, for each node j included in the list, add the impedance of link i j to the tree tables current total impedance the node i. This quantity represents a total impedance to node j via node i. It is simply represents or explains the process of proceeding further and then identifying the nearest node and considering that node as node i to proceed further. If this value is smaller than the current tree table entry for node j, the current total impedance to j with the new total impedance and enter node i as a node that immediately proceeds j.

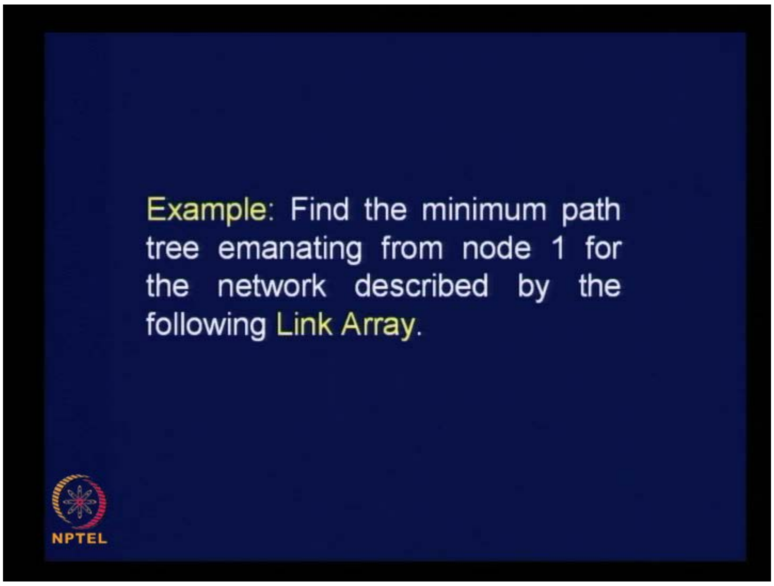
I should repeat to make things little clear, if this value is smaller than the current tree table entry for node j; node j can be accessed by different alternate routes. Let say we are got some value already as a cumulative travel time for j and we are again calculating the cumulative travel time by another route to the same point j. If this latest calculated value is smaller than the current or already calculated tree table entry for node j replace a current total obviously, because it is smaller impedance to j with the new total impedance and enter node i as a node that immediately precedes j.

(Refer Slide Time: 19:08)



To proceed further, you just name that node as node i. This operation replaces the longer path to node j with the shorter one just discovered. If the new total impedance is greater than the current tree table entry, proceed to the next link in the list. Return to step 2, unless the list is empty in which case the tree table contains the solution. Unless the list is exhausted you have to continue the process. I hope all of you have some vague idea of the steps involve in the process. We will just take numerical example in see how these steps can be followed 1 by 1 to get the minimum path tree.

(Refer Slide Time: 20:00)



To make the job of understanding easier, we will consider the same link array table about which we had a discussion the previous class. Find the minimum path tree emanating from node 1 for the network described by the following link array table. its nothing but, the table that we have already seen.

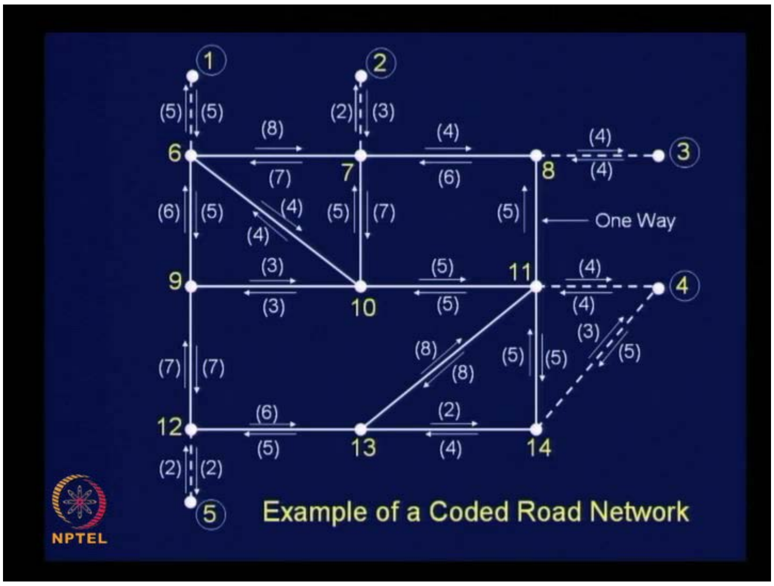
(Refer Slide Time: 20:20)

Link Array

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1						5								
2							3							
3								4						
4											4			5
5												3		
6	5						8		5	4				
7		2				7		4		7				
8			4				6							
9						6				3		7		
10						4	5		3		5			
11				4				5		5			8	5
12					2				7				6	
13											8	5		2
14				3							5		4	

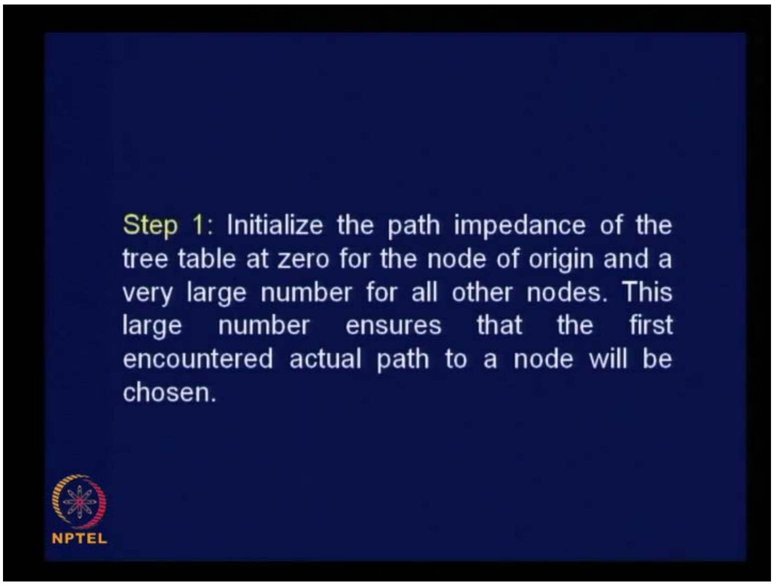
It is nothing but, the table that we have already seen. There are 14 nodes in the network and nodes 1 2 3 4 and 5 are zone centroids and other nodes are road intersections. To perceive this link array table better, I will just show you the picture also.

(Refer Slide Time: 20:46)



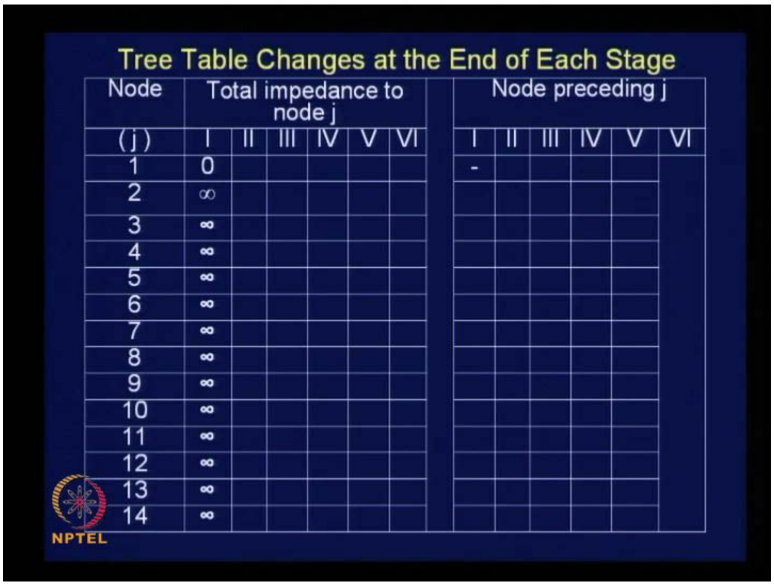
This is the network that, we are talking about nodes 1 2 3 and 4 as well as 5 are zone centroids. Other nodes are road intersections and there are two dummy links for zone centroid 4 and all other centroids have only 1 dummy link. Please try to capture this picture to understand the subsequent discussion clearly. Now our starting node is node 1 so we are trying to build minimum path tree for node 1, which is zone centroid 1.

(Refer Slide Time: 21:44)



Let us again, recapitulate step 1 we are initializing the path impedance of the tree table at zero for the node of origin which is node 1 and a very large number for all other nodes. This large number ensures that the first encountered actual path to a node will be chosen.

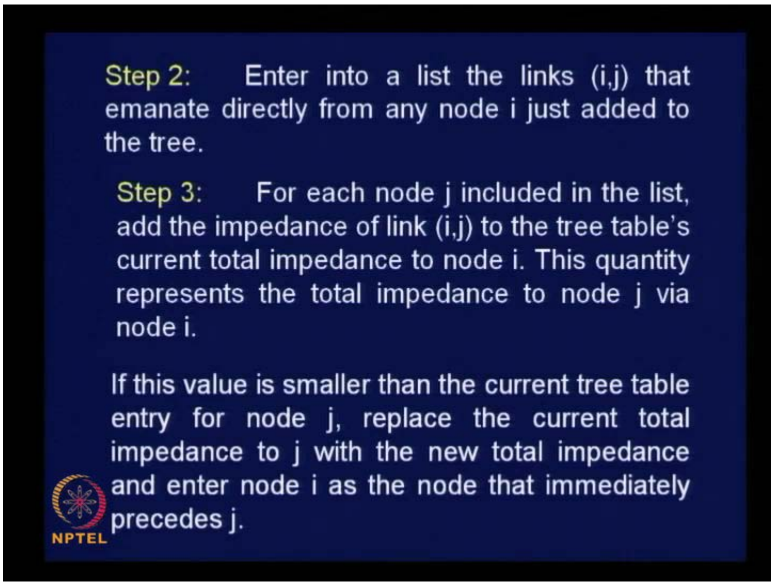
(Refer Slide Time: 22:13)


Tree Table Changes at the End of Each Stage

Node	Total impedance to node j						Node preceding j					
(j)	I	II	III	IV	V	VI	I	II	III	IV	V	VI
1	0						-					
2	∞											
3	∞											
4	∞											
5	∞											
6	∞											
7	∞											
8	∞											
9	∞											
10	∞											
11	∞											
12	∞											
13	∞											
14	∞											

Once you complete this first step, this will be the result. We have listed all the 14 nodes and the impedance to node 1 is indicated as zero. The travel time to reach out node 1 is given as zero than means; we are starting from node 1 itself and the travel time to reach out to all the other nodes as been given simply as infinity very large number. We anticipate some steps like total impedance to node j in step 1 step 2 3 4 5 6 etcetera that is why they got a comprehensive table prepared for the purpose. Similarly, we are interested to know the node preceding j, so that the link connectivity is clearly too known to us. So, that has to be entered for every step or every step of assigning your traffic.

(Refer Slide Time: 23:19)



Next, enter into a list the links i j that emanate directly from any node i just added to the tree. In this case, we can treat i as node 1 and we are interested in i j all the nodes connected to node 1, which node is connected to node 1. Node 6, if you recall node 6 is directly connected to node 1. Then for each node j included in the list we would have included node 6 in the list add the impedance of link i j. We corresponding travel time to reach 6 from 1 has to be entered there to the tree tables current total impedance to node i. This quantity represents a total impedance to node j via node i. As of now i is 1; j is 6.

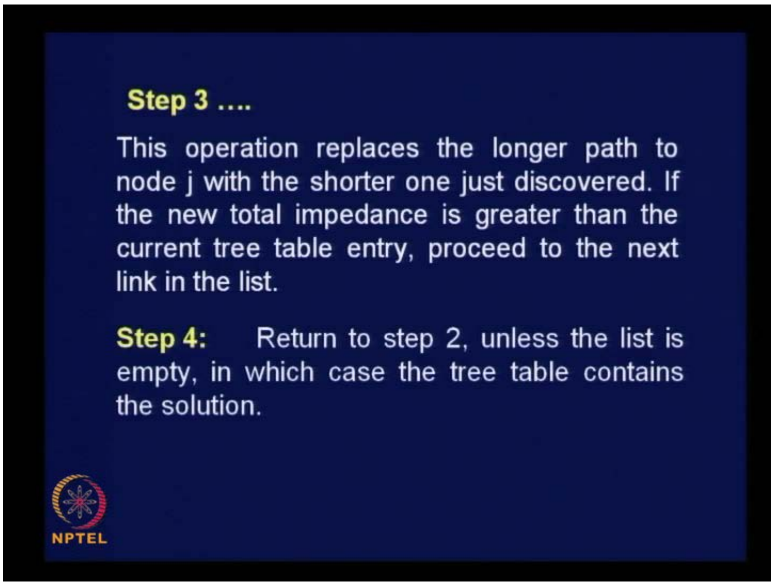
So, that impedance we can easily enter in the corresponding cell of the table. If this value is smaller than the current tree table entry for node j, node j is 6 for 6 we have entered a value of infinity obviously, the first step it is going to be much less than the current value of infinity, replace a current total impedance to j with the new total impedance and enter node i as node that immediately proceeds j.

(Refer Slide Time: 25:16)

Stage N	Links i	j	Compute new path impedence	Compare to tree table stage N-1	Decision
II	1	6	0 + 5 = 5	5 < ∞	Accept

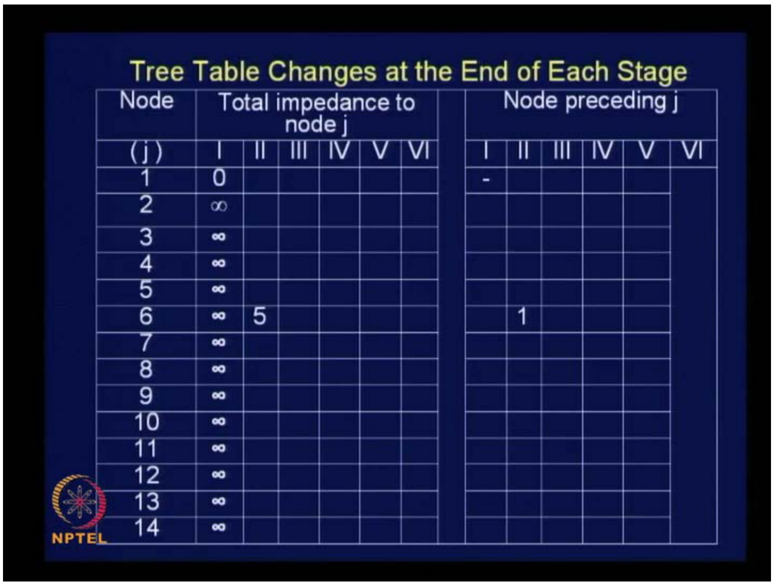
So this is the result that, we get on completion of this steps that we discussed. Stage 1 is completed now we are in stage 2. In stage 2 link i, link i j has starting node of i as 1 ending node is 6 and compute new path impedance. We are computed zero plus 5 is 5 minutes and then compared to tree table stage n minus 1, the previous stage which was infinity. So, we say 5 minutes is less than infinity so, we accept this travel time as 5 minutes. 1 to 6 is 5 minutes even though, it is obvious we take the step, because this is the step to be followed for a large network. Since our network very very small we feel that it is very elementary, where this procedure has to be followed to cover thousands of links and nodes.

(Refer Slide Time: 26:31)



Then go to the step 3. This operation replaces the longer path to mode j with the shorter one just discovered. Of course, this is not there in the first step itself, because we have not come to that stage. If new total impedance is greater than the current tree table entry, proceed to the next link in the list. Then, step 4 returns to step 2 unless a list is empty in which case the tree table contains the solution.

(Refer Slide Time: 27:06)

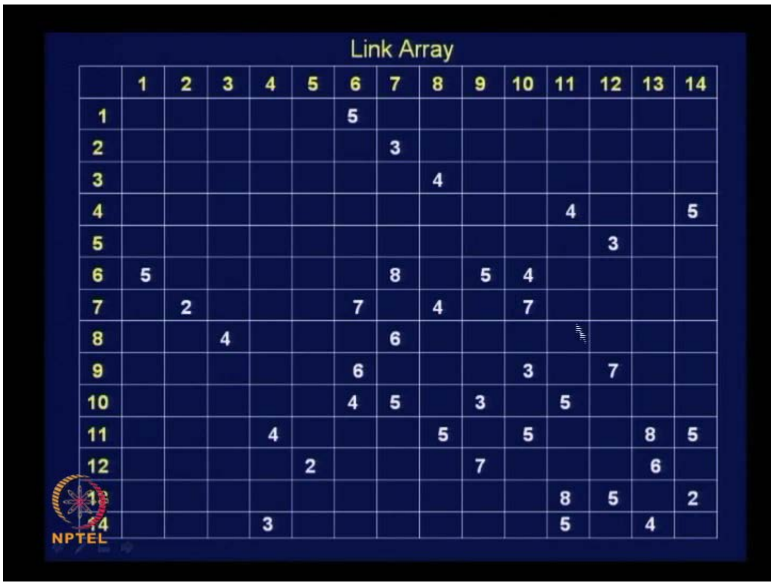


Tree Table Changes at the End of Each Stage

Node	Total impedance to node j						Node preceding j					
(j)	I	II	III	IV	V	VI	I	II	III	IV	V	VI
1	0						-					
2	∞											
3	∞											
4	∞											
5	∞											
6	∞	5						1				
7	∞											
8	∞											
9	∞											
10	∞											
11	∞											
12	∞											
13	∞											
14	∞											

So, this is the result that we get on completion of step 2. Are you able to appreciate this point? Node j of our interest is node 6 and to reach out to this node from our node of interest the cumulative travel time is 5 minutes 1 2 6 and for this node a preceding node is 1. We have started from 1 so node preceding j is 1, node j is 6 and travel time cumulative is 5. We will go to the next step.

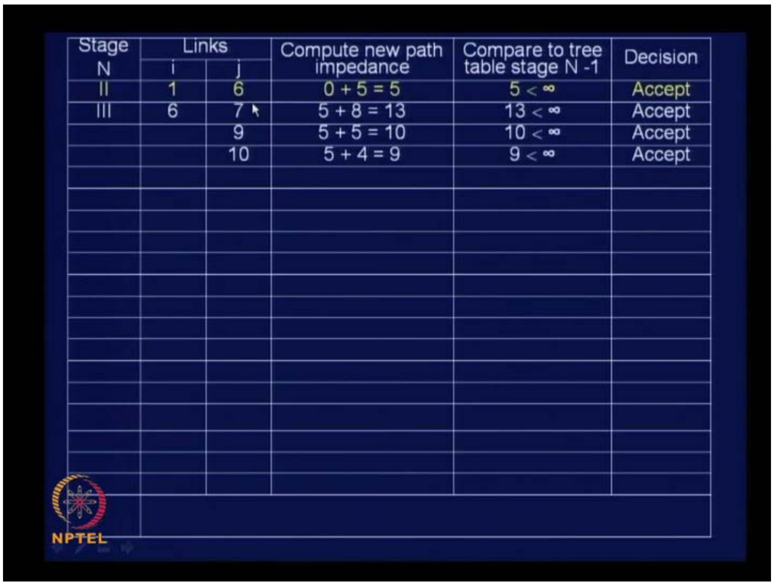
(Refer Slide Time: 28:04)


Link Array

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1						5								
2							3							
3								4						
4											4			5
5												3		
6	5						8		5	4				
7		2				7		4		7				
8			4				6							
9						6				3		7		
10						4	5		3		5			
11				4				5		5			8	5
12					2				7				6	
13											8	5		2
14				3							5		4	

Now, have a look at the link array. We have to proceed from node 6. We have come from 1 to 6 and proceed further, from 6 to other nodes and please note 6 is connected to apart from 1 the nodes 7 9 and 10. Node 7 9 and 10 are connected to 6.

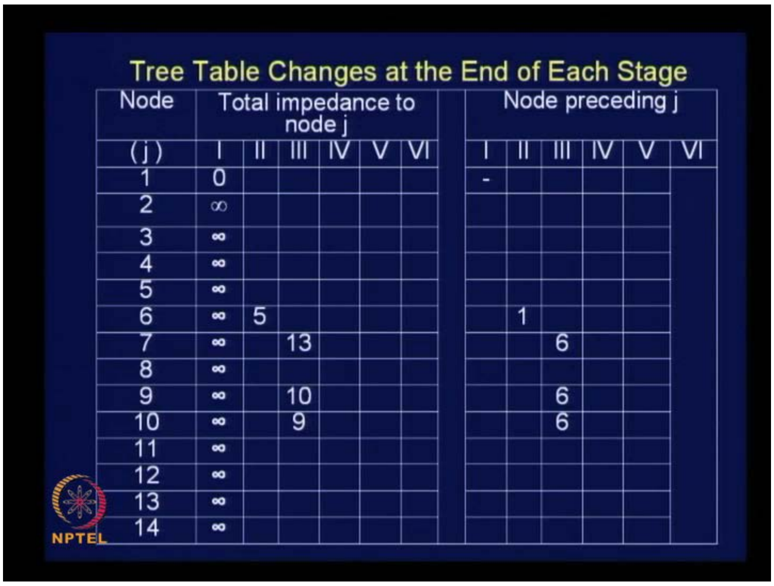
(Refer Slide Time: 28:46)



Stage N	Links i	Links j	Compute new path impedance	Compare to tree table stage N -1	Decision
II	1	6	0 + 5 = 5	5 < ∞	Accept
III	6	7	5 + 8 = 13	13 < ∞	Accept
		9	5 + 5 = 10	10 < ∞	Accept
		10	5 + 4 = 9	9 < ∞	Accept

So, let us proceed towards these three nodes from 6 following the same algorithm. If you do so, your i becomes 6 now. We are proceeding from 6, whereas in earlier iteration 6 was j now it is i and j nodes are 7 9 and 10 as we identify. The cumulative travel time from 1 to reach 7 through 6 is 5 plus 8 13 minutes and cumulative travel time to reach 9 from 1 through 6 is 5 plus 5 10 minutes and the cumulative travel time to reach 10 is 5 plus 4 9 minutes. Then, check whether these current values are less than the values that we have assigned or those we have obtained for these nodes in the previous iteration and previously we have assigned the values of infinity only. So we find that all these numbers are less than infinity, we accept these things as the cumulative shorter path from node 1.

(Refer Slide Time: 30:12)

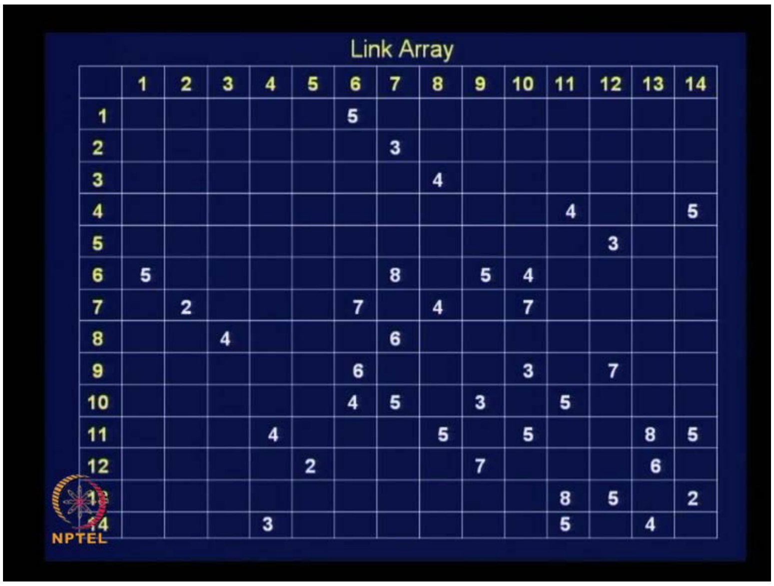


Tree Table Changes at the End of Each Stage

Node	Total impedance to node j						Node preceding j					
(j)	I	II	III	IV	V	VI	I	II	III	IV	V	VI
1	0						-					
2	∞											
3	∞											
4	∞											
5	∞											
6	∞	5						1				
7	∞		13						6			
8	∞											
9	∞		10						6			
10	∞		9						6			
11	∞											
12	∞											
13	∞											
14	∞											

So, obviously we will be proceeding further from these nodes and of course, this is the result we are updating this table to reflect the changes that we have made. Now to reach out the 7 total impedance to node j j is 7 13 minutes and for 9 10 minutes for 10 9 minutes, the node preceding 7 is 6, preceding 10 is also 6, preceding 9 is also 6 and preceding 10 is also 6, preceding nodes are also written down. These are necessary to finally, get the minimum path that is idea of writing down a preceding nodes also otherwise you will not be able to get the link connectivity.

(Refer Slide Time: 30:57)



Link Array

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1						5								
2							3							
3								4						
4											4			5
5												3		
6	5						8		5	4				
7		2				7		4		7				
8			4				6							
9						6				3		7		
10						4	5		3		5			
11				4				5		5			8	5
12					2				7				6	
13											8	5		2
14				3							5		4	

Now, go back to the link array. We have to proceed from 7 9 and 10. Look at the connectivity to these nodes. 7 are connected to 2 of course, apart from 6 we have already seen 6 to 7 we have come. The new connectivity is 7 to 2 7 to 8 and 7 to 10. If you take node 9 it is connected to 6, which is already known to us we have already done. Then 9 is connected to 10 and 12 and node 10 apart from 6 is connected to 7 9 as well as 11. So, we have going to proceed towards these nodes from each of the three identified nodes all these three nodes will become i, now and we are going to proceed towards j this is what we have seen what we need to do is this.

(Refer Slide Time: 32:13)

Stage N	Links i	Links j	Compute new path impedence	Compare to tree table stage N-1	Decision
II	1	6	0 + 5 = 5	5 < ∞	Accept
III	6	7	5 + 8 = 13	13 < ∞	Accept
		9	5 + 5 = 10	10 < ∞	Accept
		10	5 + 4 = 9	9 < ∞	Accept
IV	7	2	13 + 2 = 15	15 < ∞	Accept
		8	13 + 4 = 17	17 < ∞	Accept
		10	13 + 7 = 20 Reject		Reject
	9	10	10 + 3 = 13	13 > 9	Reject
		12	10 + 7 = 17	17 < ∞	Accept
	10	7	9 + 5 = 14	14 > 13	Reject
		9	9 + 3 = 12	12 > 10	Reject
		11	9 + 5 = 14	14 < ∞	Accept

Now, 7 to 2 to reach out to 7 the cumulative travel time is 13, we have already seen. 13 plus the additional link time link travel time is 2. So, total time is 15 and this is less than the initial assigned value of infinity, so we are accepting this. As the cumulative travel time from 1 to this node and 7 to 8 13 plus 4 17, this is also less than infinity so we are accepting. Then 7 to 10 13 plus 7 20 check whether, we have reached out 10 with a lesser travel time. You can see that we have already reached out 10 with a travel time of 9 minutes. 1 6 10 is 9 minutes whereas, now we get a travel time of 20 minutes to reach 10. Obviously, it is more than the previously established value for this particular node, so we are rejecting. Then 9 to 10 plus 3 13 minutes and 9 to of course, it is greater than 9, so we are rejecting. Please, remember we are reaching node 10 within 9 minutes as per the earlier calculations there is no need to accept this we can reject.

9 to 12 is 7 minutes 10 plus 7 17 is the cumulative travel time and to reach out to 12 17 as of now is shortest time is less than infinity we accept it and then 10 to 7 9 plus 5 14 is greater than 13. We are able reach node 7 within 13 minutes, so we are rejecting this path and 10 to 9 9 plus 3 12 is greater than 1 again we have to reject this is not lying in the shortest path and 10 to 11 9 plus 5 14 to reach out to 11 14 is the shortest time as of now less than infinity we are accepting. All the rejected links will not appear in your minimum path tree that is the point to be understood that clear. Now what are the links that we are what are the nodes we have reached out now. Better go back and have a look so, where from to proceed now. That we have to see what the cases we are accepted are. We have accepted this node as the next 1 and this 8 is also accepted, 10 are rejected and then 12 are accepted and from 10 we accept 11.

This implies, when you proceed from proceed further we have to proceed from 2 as well as 8 then from 12 and then from 11. So, there are 4 nodes they were from we need to proceed further. The related question is can we proceed further with 2 is zone centroid that means we have reached end point. One to 2 for this complete there is no need to worry about the extension from 2, the point of interest is already reached. So, 2 need not be considered actually so we need to consider only 8 then 11 and then 12. So, these are the nodes were from we need to proceed further.

(Refer Slide Time: 36:54)


Link Array

	1	2	3	4	5	6	7	8	9	10	11	12	13	14
1						5								
2							3							
3								4						
4											4			5
5												3		
6	5						8		5	4				
7		2				7		4		7				
8			4				6							
9						6				3		7		
10						4	5		3		5			
11				4				5		5			8	5
12					2				7				6	
13											8	5		2
14				3							5		4	

So, when you consider node 8 look at the connectivity is connected to 3 7 we already seen, its only 1 connection we need to consider 8 to 3 needs to be considered. Then 11 to 4 11 to 8 and 11 to 10 11 10 connection, how we have considered then we will be interested in 11 4 and 11 5. As well as 11 13 and 11 14 these are the connecting nodes then, if you consider 12 5 9 and then 13.

(Refer Slide Time: 38:07)

Stage N	Links i	Links j	Compute new path impedence	Compare to tree table stage N-1	Decision
II	1	6	0 + 5 = 5	5 < ∞	Accept
III	6	7	5 + 8 = 13	13 < ∞	Accept
		9	5 + 5 = 10	10 < ∞	Accept
		10	5 + 4 = 9	9 < ∞	Accept
IV	7	2	13 + 2 = 15	15 < ∞	Accept
		8	13 + 4 = 17	17 < ∞	Accept
		10	13 + 7 = 20 Reject		Reject
	9	10	10 + 3 = 13	13 > 9	Reject
		12	10 + 7 = 17	17 < ∞	Accept
	10	7	9 + 5 = 14	14 > 13	Reject
		9	9 + 3 = 12	12 > 10	Reject
		11	9 + 5 = 14	14 < ∞	Accept
V	8	3	17 + 4 = 21	21 < ∞	Accept
	11	4	14 + 4 = 18	18 < ∞	Accept
		8	14 + 5 = 19	19 > 17	Reject
		13	14 + 8 = 22	22 < ∞	Accept
		14	14 + 5 = 19	19 < ∞	Accept
	12	5	17 + 2 = 19	19 < ∞	Accept
		13	17 + 6 = 23 Reject		Reject

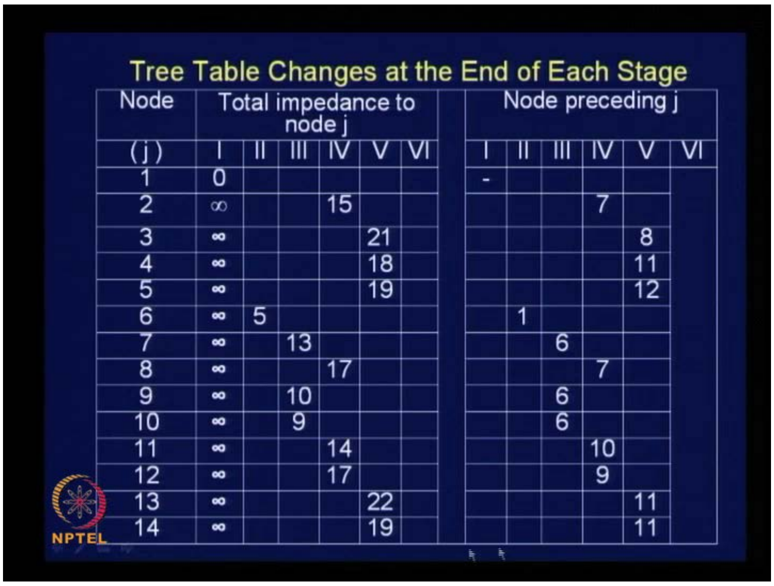
So, while looking at this link array you should be able to perceive the picture. That is the most important aspect and then proceeds further. So, from 8 we proceed to 3 we cumulative travel time already to 8 is 17 and 8 to 3 travel time is 4 the cumulative time is 20 1 to reach out to 4, it is less than infinity so we are accepting. Also please remember 4 this node 3 is another zone centroid. We are reaching the end point of interest in this particular case and then 11 to 4 14 plus 4 18 and to reach out to 4 this is this happens to be the least travel time, as of now 18 is less than infinity we are accepting. Then, 11 to 8 14 plus 5 19 but, 19 is more than 17. Because, we have shown earlier that we can reach 8 with the cumulative travel time of 17.

Here, we can reach out to 8 with the cumulative time of 17 minutes so we rejected. Then 11 to 13 14 plus 8 22, 22 is less than infinity so we accepted and then reach 14 from 11 14 plus 5 19 is less than infinity we accept and then to reach 5 from 11 17 plus 2 9 less than infinity we are accepting these are all new paths.

And 12 to 5 is what we saw is 17 plus 2 19 less than infinity we accept and then 12 to 13, 13 plus 6 23 we are rejecting. Why? Please, it is an interesting case what we have given as title for this column is this compare to tree table stage n minus 1, n minus 1 is stage 4 actually. Normally, we compare to reject the values obtain in the previous step, whereas in this case its very interesting to note that within this step we have reached out to this node 13 and got a value less than this value. That is why, I have not written anything in this column I have just written reject here itself. Because, it is not as per the guide line given on top of this particular column. It is in comparison with the previous step, where as what they do here is in comparison within this step. Anyway we have to reject, so I write reject here and reject in the final decision making column also.

Where do we stand now? So, this is end point there is no need to proceed further from 3 and what are the things we have accepted. There is scope to proceed further from 4 because, we have accepted and there is no way to proceed further from 8 and we can proceed further from 13. Because, we are accepted and we can also proceed further from 14 and we can also proceed from 5 but, there is no way to proceed from 13 here of course, here these is scope. So the nodes where from we will proceed further are excellent that is the point we have to make, I am happy that all of you are alert. So, 3 4 and 5 are zone centroids there is no need to proceed further. So, the nodes where from we need to proceed further or only 13 and 14 only two nodes.

(Refer Slide Time: 42:40)


Tree Table Changes at the End of Each Stage

Node	Total impedance to node j						Node preceding j					
(j)	I	II	III	IV	V	VI	I	II	III	IV	V	VI
1	0						-					
2	∞			15						7		
3	∞				21						8	
4	∞				18						11	
5	∞				19						12	
6	∞	5						1				
7	∞		13						6			
8	∞			17						7		
9	∞		10						6			
10	∞		9						6			
11	∞			14						10		
12	∞			17						9		
13	∞				22						11	
14	∞				19						11	

Of course, this is the finished table to reflect the current situation. We are simply listing j and the corresponding total impedance and the node preceding j. This is just help us to understand, where we are with respect to cumulative travel time and the minimum path tree.

(Refer Slide Time: 43:07)

Stage	Links		Compute new path impedence	Compare to tree table stage N-1	Decision
N	i	j			
II	1	6	0 + 5 = 5	$5 < \infty$	Accept
III	6	7	5 + 8 = 13	$13 < \infty$	Accept
		9	5 + 5 = 10	$10 < \infty$	Accept
		10	5 + 4 = 9	$9 < \infty$	Accept
IV	7	2	13 + 2 = 15	$15 < \infty$	Accept
		8	13 + 4 = 17	$17 < \infty$	Accept
		10	13 + 7 = 20 Reject		Reject
	9	10	10 + 3 = 13	$13 > 9$	Reject
		12	10 + 7 = 17	$17 < \infty$	Accept
	10	7	9 + 5 = 14	$14 > 13$	Reject
		9	9 + 3 = 12	$12 > 10$	Reject
		11	9 + 5 = 14	$14 < \infty$	Accept
V	8	3	17 + 4 = 21	$21 < \infty$	Accept
	11	4	14 + 4 = 18	$18 < \infty$	Accept
		8	14 + 5 = 19	$19 > 17$	Reject
		13	14 + 8 = 22	$22 < \infty$	Accept
		14	14 + 5 = 19	$19 < \infty$	Accept
	12	5	17 + 2 = 19	$19 < \infty$	Accept
		13	17 + 6 = 23 Reject		Reject
	All the links emanating from nodes 3, 4, 5, 13 and 14 are rejected; the list is now empty and the procedure ends.				

Now, we are proceeding from we decided to proceed from these nodes and we find those nodes are n nodes. So, there is no way to proceed further. Our remark is this all the links emanating from nodes 3 4 5 13 and 14 are rejected, the list is now empty and the procedure ends. So, we accept whatever we have done so far and so this is the final value as per the tabulation, we have initially made we have information about the cumulative travel time as well as the connecting nodes.

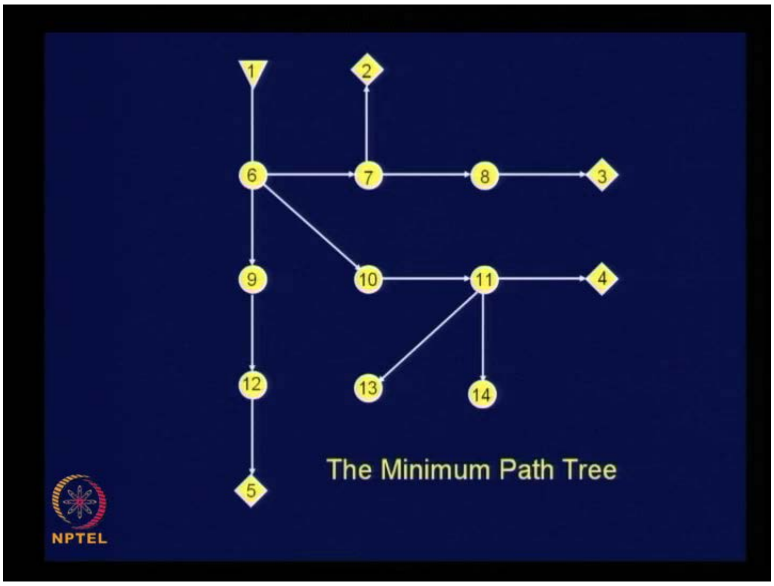
(Refer Slide Time: 44:16)

Final Tree Table

Node (j)	Total impedance to node j	Node preceding j
1	0	-
2	15	7
3	20	8
4	18	11
5	19	12
6	5	1
7	13	6
8	17	7
9	10	6
10	9	6
11	14	10
12	17	9
13	22	11
14	19	11

And this is the final summary of the results that, we have got. Total impedance to j and node preceding j that is what we want and using this information you can make a minimum path tree. Before that, have a look at the network that we have initially considered and the minimum path of tree is this.

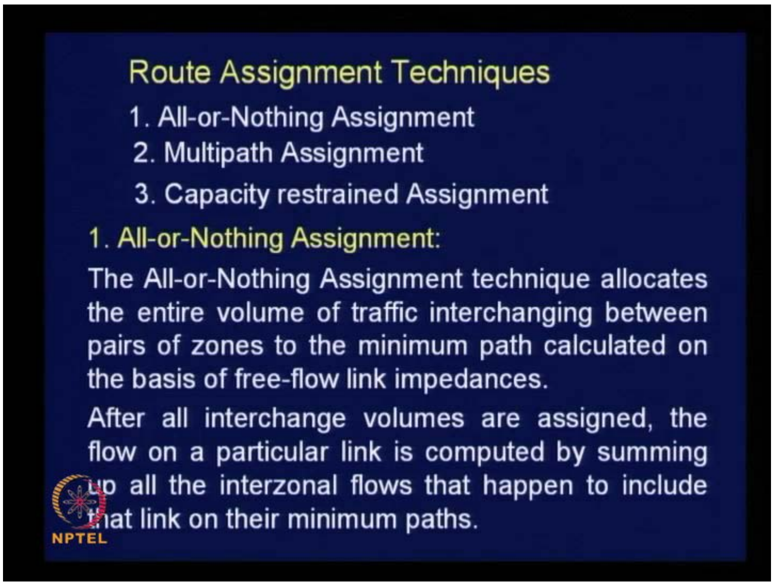
(Refer Slide Time: 44:40)



From node 1 to all the other nodes. How many links are involved? Here, 13 links are involved and the original network has how many links including the dummy links, how many because, dummy links are shown here as minimum paths. Even though they are not distinctly shown in dotted line all the dummy links are included except one. We had 13 plus 6 19 links in the original network; if you can just look at we have 13 links in the road network and 6 dummy links. So totally 19 links and 19 links have become just how many 13. So 6 links are not needed to reach out to all the nodes of interest from node 1. This is the minimum path tree. We will be assigning traffic from 1 to all the other points of interest following this route only as per user equilibrium assignment methodology.

So please understand that link array can be given as input to get minimum path tree from any node of your interest. That is a point to be understood very clearly. So your input is just the link array and you can click a button indicating your reference node, you will get this minimum path tree not in the form of picture again in the form of a table, indicating the cumulative travel time and the corresponding nodes.

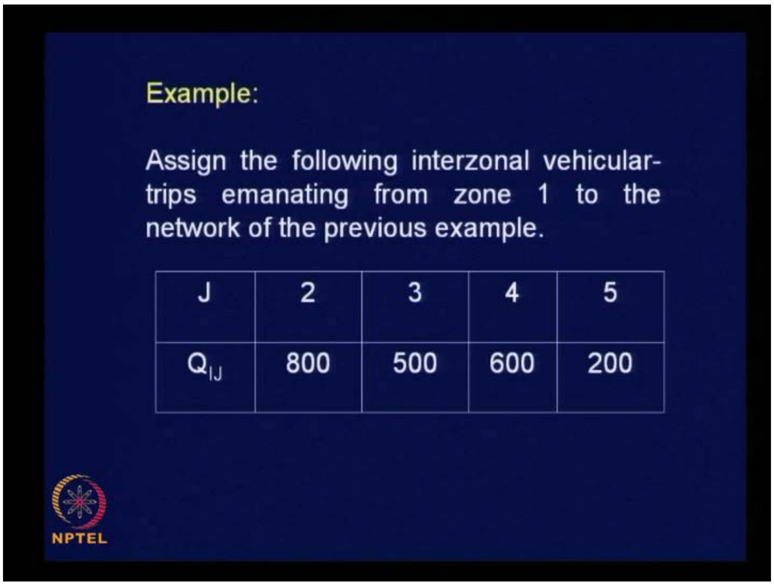
(Refer Slide Time: 46:58)



Now, let us discuss about the route assignment techniques. Even under user equilibrium principle there are different techniques available for assigning traffic. The first one is all or nothing assignment technique; second multi path assignment technique; third capacity restrained assignment technique. Basically, there are three possibilities for assigning traffic.

Let us first discuss about all or nothing assignment technique. This technique allocates the entire volume of traffic interchanging between pairs of zones to the minimum path calculated on the basis of free flow link impedances. That is why this is called all are nothing assignment technique. You assigned the whole of the traffic to a set of links are nothing to a given path. So, this has per this technique the allocation is done to the entire volume of entire volume of traffic interchanging between pairs of zones to the minimum path calculated on the basis of free flow link impedances, as I indicated to you it is free flow link impedance, that is taken as the basis for assigning traffic, after identification of minimum path tree. All the travel times that we indicated as link attributes or free flow travel times. After all interchanges volumes are assigned the flow on a particular link is computed by summing up all the inter zonal flows that happen to include that link on their minimum paths.

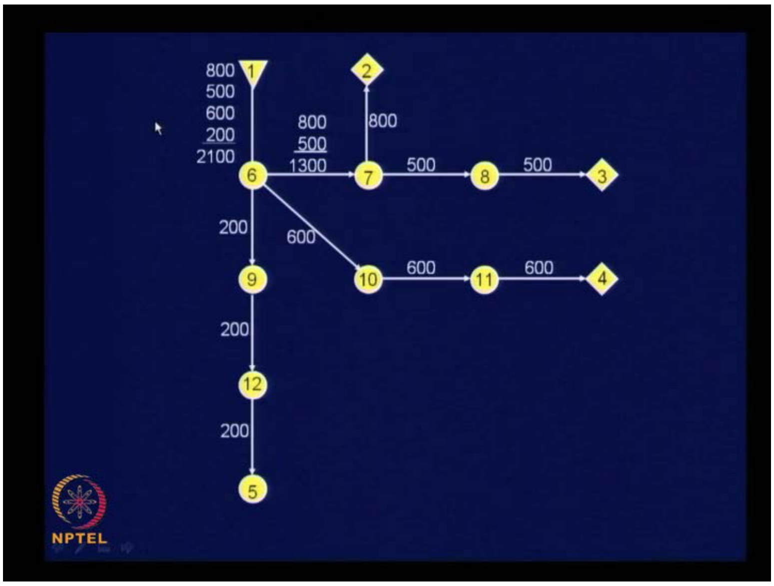
(Refer Slide Time: 49:31)



So, very simple methodology. Let us take this a small example. Assign the following inter zonal vehicular trips emanating from zone 1 to the network of the previous example. We have already built a minimum path tree. We lose a same tree and assign the traffic volume shown here from 1 to all other nodes of our interest. One is also zone centroid, we are going to assign the traffic from 1 to other zones 2 3 4 and 5. The traffic volumes are 1 to 2 is 800, 1 to 3 500, 1 to 4 600, 1 to 5 200. Normally, traffic volume is quantified in terms of p c u passenger car unit.

You may recall we discussed about conversion of person trips into vehicular trips that is 1 stage under heterogeneous traffic conditions, that too highly heterogeneous traffic conditions prevailing in our country. We cannot just measure vehicular traffic in terms of number of vehicles. Even though it may be approximately under highly homogenous traffic conditions, where around 90 percent of vehicles are cars. And to have a realistic idea about traffic volume, it is very important to convert to heterogeneous traffic different types of vehicle into equivalent passengers cars. So these numbers indicate p c u values, Otherwise you will not able to check whether the capacity of link is reached or not that can be assessed, only based on the traffic measured in terms of p c u.

(Refer Slide Time: 51:34)



So, I have just shown the result to you may recall volume of traffic to be assigned from 1 to 2 is how much 800 p c u; p c u per day or per hour what is the normal measure per day are you sure it is per hour. We talked about assigning for morning peak, evening peak and non peak hours and so on. It is hourly volume of traffic we should change our from daily traffic volume to hourly traffic volume we do route assignment. So you can see 800 p c u is loading this link that is what I have written here 800 is loading link 6 7 also and 7 2 is also loaded with 800 p c u that completes the traffic flow from 1 to 2. And 1 2 3 500 p c u that 500 also loads this link 1 6 is loaded with 500, 6 7 also is loaded with 500 because, it is lying on the shortest path 7 8 is loaded with 500 and 8 3 500.

And 1 to 4 600 p c u and 1 6 is loaded with that amount of traffic volume also 600, then 6 10 600, 10 11 600, 11 4 600 and 1 2 5 200 p c u per hour. So 1 to 6 200, 6 to 9 200, 9 to 12 200, 12 to 5 200. Now, sum up the assigned units of traffic volume for each of the links we have 4 components for 1 6 totaling to 2100 p c u per hour and there are 2 components for 6 7 amounting to 1300, all other links have only one component. Now the picture with regard to link volumes is very clear to us. So this is how we assigned traffic as per all or nothing assignment technique. We have assigned all the traffic along the shortest path that is why the minimum path tree is used the we assigned nothing to the other alternative paths to reach out to these nodes from one that is how we just named this technique has all or nothing assignment technique.

So, we will stop here to summarize what we have seen today, you may recall we started our discussion with understanding the procedure for building minimum path tree right. We took a small numerical example, and understood the basic principle involved in development of minimum path tree, and that was just for the purpose of understanding the principle in reality, we need to make use of link array table for development of minimum path tree. And we went through the four important steps related to use of link array for developing minimum path tree, and we completed the process and understood how computationally the development of minimum path tree can be managed using link array table.

Then, we discussed about the three techniques available for route assignment, namely all or nothing assignment technique, multi path assignment technique, and capacity restrained assignment technique. We took a numerical example to understand better the all or nothing assignment technique, we will proceed the rest of it in the next class.