

COMPUTER NETWORKS
Prof. Sujoy Ghosh
Department of Computer Science and Engineering
Indian Institute of Technology, Kharagpur
Lecture-27
RIP- Distance Vector Routing

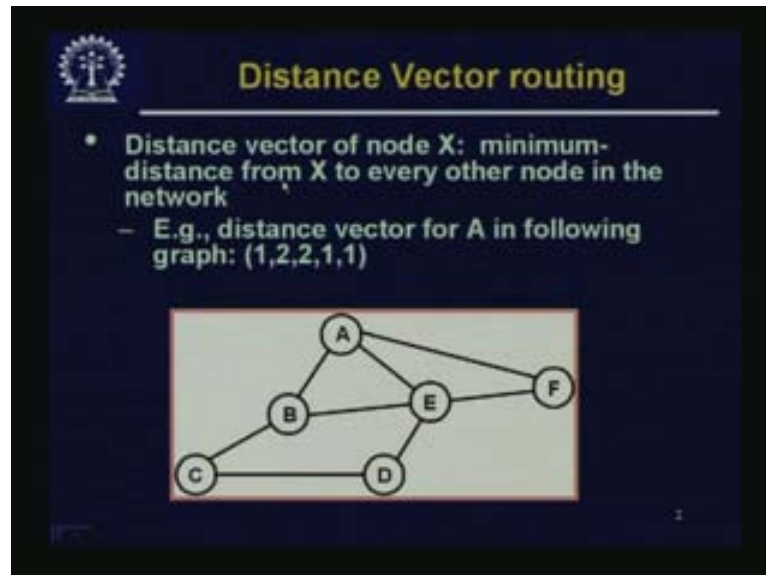
We have seen basic routing. Now we will go to some important and specific routing protocols. The first routing protocol we will be discussing today is Distance Vector Routing or RIP routing information protocol.

(Refer Slide Time: 01:08 to 01:25min)



RIP is one of the oldest routing protocol actually used in the internet and is still used in many places which is based on distance vector.

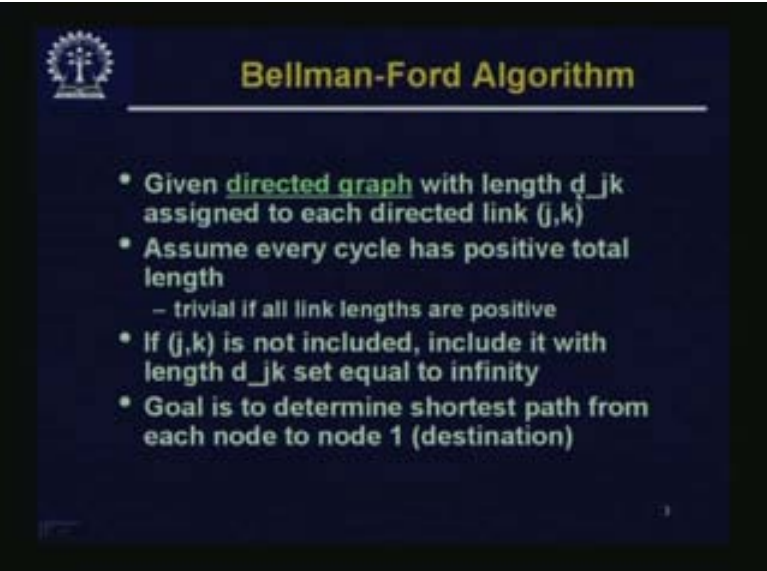
(Refer Slide Time: 01:25 to 03:40min)



The distance vector of node X is the minimum distance from X to every other node in the network. For example, the distance vector for A in the following graph is 1, 2, 2, 1, 1 that means for B it is the distance one, for C it is 2, for D it is the distance 2, for E and F it is the distance 1. So from A these are the distance vectors. If you recall from the last lecture, the routing problem can be reduced to a graph theoretic problem so these nodes are the routers but there could be some hosts also.

But for our purpose of RIP we will consider these to be routers. When we are not talking about delay, congestion, link, capacity etc in the simplest case this is the shortest path problem. So the distance here is just the number of hops. These links are also not weighted so we just count the number of links to reach from the source to destination and that is given as the distance vector for the destination node. From the source node we will use these distance vectors to compute the shortest path. We have to run a standard shortest path algorithm and here in this particular case we will be using Bellman Ford algorithm. We have to run them in a distributed fashion. This Bellman Ford algorithm should be running at each node.

(Refer Slide Time: 03:40 to 04:25min)

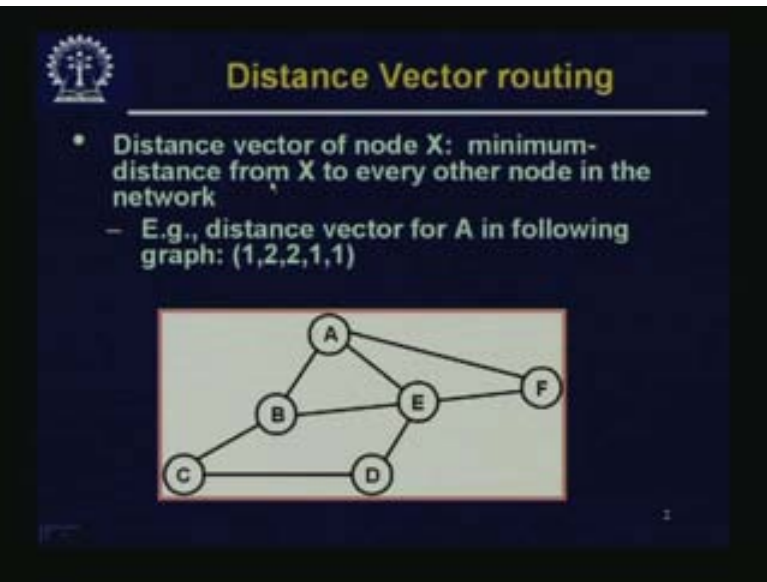


Bellman-Ford Algorithm

- Given **directed graph** with length d_{jk} assigned to each directed link (j,k)
- Assume every cycle has positive total length
 - trivial if all link lengths are positive
- If (j,k) is not included, include it with length d_{jk} set equal to infinity
- Goal is to determine shortest path from each node to node 1 (destination)

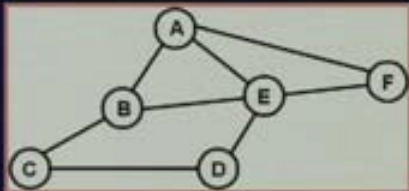
As per the Bellman Ford algorithm given directed graph with length d_{jk} which means this is the distance vector from j to k assigned to each directed link $j \rightarrow k$. Assume every cycle has positive total length. This of course is trivial, if all the links are positive and since we are taking the link weight to be 1 this will always be valid. If (j,k) is not included then include it with length d_{jk} set is equal to infinity. The goal is to determine the shortest path from each node to node 1 which is the destination.

(Refer Slide Time: 04:24 to 04:27 min)



Distance Vector routing

- Distance vector of node X : minimum-distance from X to every other node in the network
 - E.g., distance vector for A in following graph: (1,2,2,1,1)



(Refer Slide Time: 04:27 to 04:51min)



Bellman-Ford Algorithm

- Given **directed graph** with length d_{jk} assigned to each directed link (j,k)
- Assume every cycle has positive total length
 - trivial if all link lengths are positive
- If (j,k) is not included, include it with length d_{jk} set equal to infinity
- Goal is to determine shortest path from each node to node 1 (destination)

3

So this is the graph and what is this infinity? Well, the point is, when you are starting the algorithm j may not know that it is actually connected to k through multiple hops. So to start with j will just set it equal to infinity.

(Refer Slide Time: 04:51 to 07:35 min)



Bellman-Ford, cont.

- Let D_i^h be length of shortest directed path with at most h arcs from node i to destination
- D_i^h is generated by iteration

$$D_i^{h+1} = \min_j \{d_{ij} + D_j^h\}, \quad D_i^0 = \infty$$

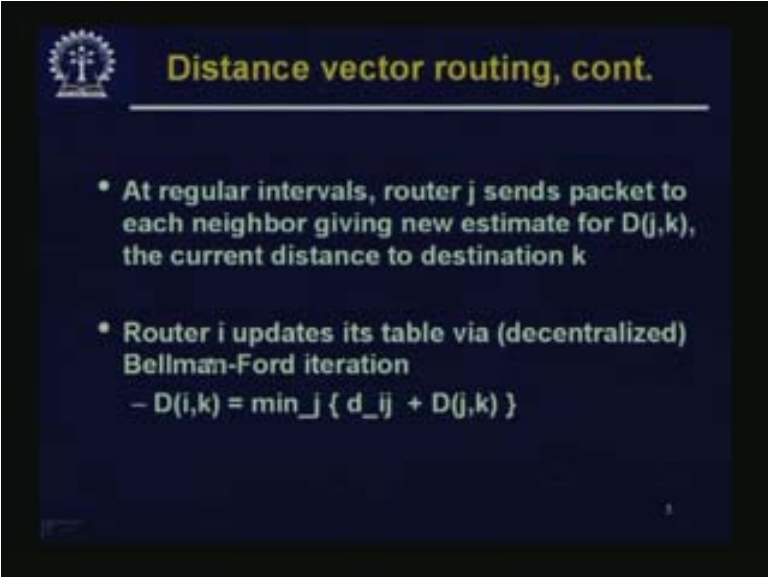
Since every cycle has positive length, no optimal path contains a cycle; hence has at most $N-1$ arcs, where N is total number of nodes. Thus, at most N iterations.

4

Let D_i^h be the length of shortest directed path with at most h arcs from node i to destination. D_i^h is generated by the iteration $D_i^h = \min_j \{d_{ij} + D_j^{h-1}\}$ and D_i^0 is equal to infinity. So you can put all of them as infinity or the direct links you can take note of them and in one iteration that will be taken care of. The D is telling us that this is the best path I know at the moment from myself to this particular node at this iteration.

Now, in another iteration when these distance vectors are interchanged we may find that my neighbor knows something which is of a better cost. So I can go to my neighbor using d_i to the power j over all j and then d_j to the power h . If there are n nodes I need to do this iteration where at most there will be n minus 1 arcs. This is the worst case in which all the nodes are connected in a chain. So, after n minus 1 iterations I will get to know about a node which is at a distance n minus 1 from me because at each iteration this information will filter through and then reach me. This node might have thought that node is not connected at all so D_i^0 is infinity. But after n minus 2 iterations its neighbor would know that it is connected to that with a distance of n minus 2 and then this information will reach this node and this will know that it can go to its neighbor in one hop and take n minus 2 hops to destination so it can reach that node in n minus 1 hops. Since every cycle has positive length no optimal path contains a cycle hence it has at most N minus 1 arcs where N is the total number of nodes. Thus it is at most N iterations including the first one.

(Refer Slide Time: 07:35 to 09:26 min)



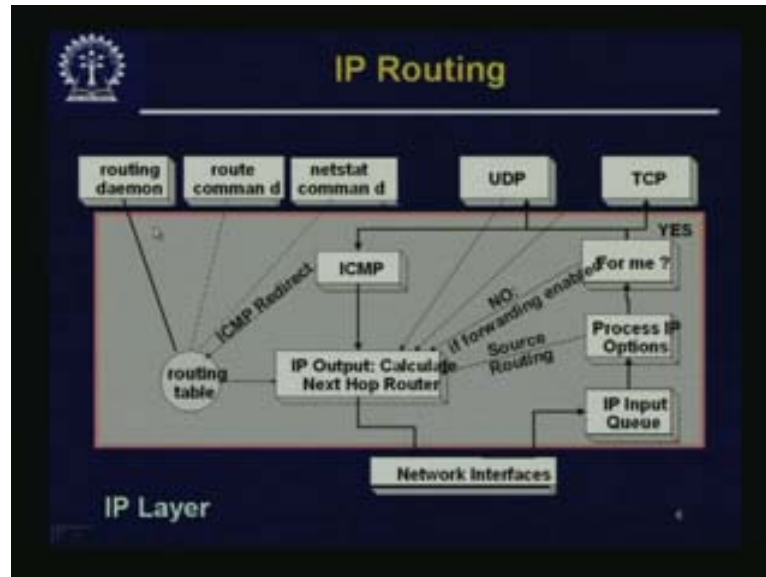
Distance vector routing, cont.

- At regular intervals, router j sends packet to each neighbor giving new estimate for $D(j,k)$, the current distance to destination k
- Router i updates its table via (decentralized) Bellman-Ford iteration
 - $D(i,k) = \min_j \{ d_{ij} + D(j,k) \}$

What we do in this distance vector routing is that at regular intervals router j sends packet to each neighbor giving new estimation for d_j to the power k the current distance to destination k . That means the router j is giving the d_j to the power k value and not only d_j to the power k value for all k so that is why this is not just one distance value but this is called a distance vector that is sent to all its neighbors. So the neighbors tell each other about the distances they know to a particular node k and update the shortest one in its table. If node j finds that SP for the node 1 from the neighbor is shorter than it own, it will update the new path. We do some local exchange of information between any router and its neighbors as the algorithm is of course running globally

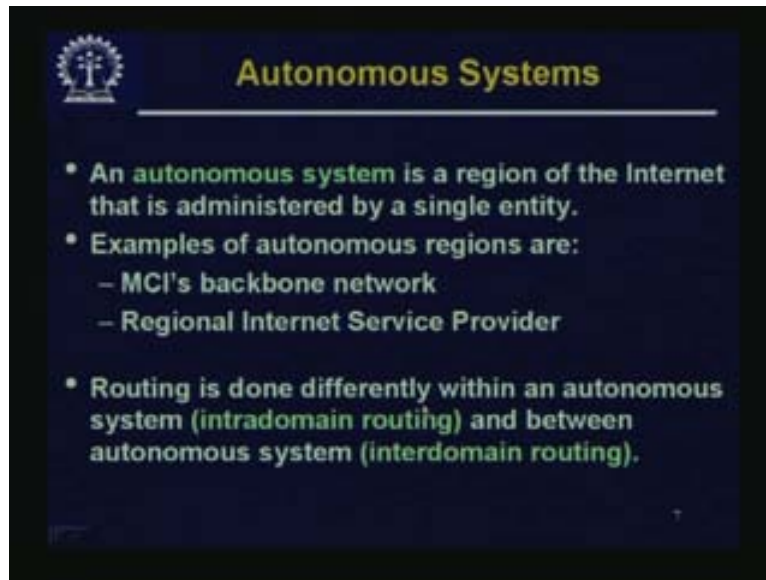
and in a distributed fashion. Router I updates its table via decentralized Bellman Ford iteration. So $D(j,k)$ is equal to minimum $\{d_i \text{ to the power } j \text{ plus } D(j,k)\}$ which gives the minimum distance to a particular destination k.

(Refer Slide Time: 09:27 to 10:55min)



For IP routing, a routing table has to be created and which has to be kept updated. There may be ICMP redirect messages from other routers make changes to this routing table. There is the routing daemon. What the routing daemon will do is that from time to time it will run and send packets to the neighboring routers or run some iterations of this Bellman Ford algorithm and keep the routing table up to date. So, from this routing table we get the IP output and calculate the next hop router and if it is source routing we get it otherwise we get it from the routing table. When a particular packet arrives it sees whether it is for the router and if yes of course it goes up, if it is no then it calculates the next hop.

(Refer time slide 10:56 to 13:47min)

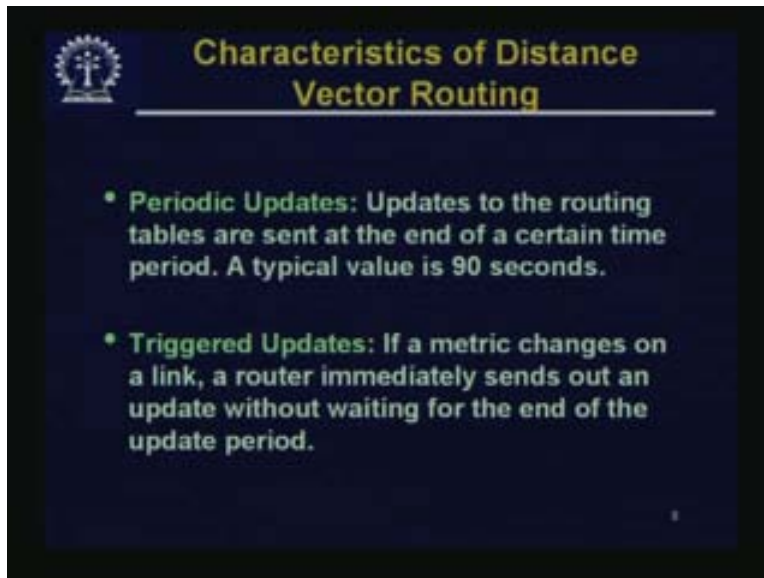


Autonomous systems are not directly used in RIP but it is used in the other algorithm. This is very important for scaling the network. We know that whatever routing protocol we are following this has to scale to the global level for millions of routers which are connected to each other. The router may be connected to a network. The number of routers together may form an autonomous system. Here we are introducing another level of hierarchy.

Autonomous system is a region of the internet that is administered by a single entity, Therefore this is not just one network but a collection of networks. Some examples of autonomous regions are service providers, MCI's with its own backbone system or Regional Internet service providers. The regional internet service providers are giving the services to so many customers and each customer has their own router connecting to this backbone. Each of the customers will have his own network so it is a collection of networks in a way. But this is one autonomous system. The idea is to run the same kind of protocol inside this autonomous system.

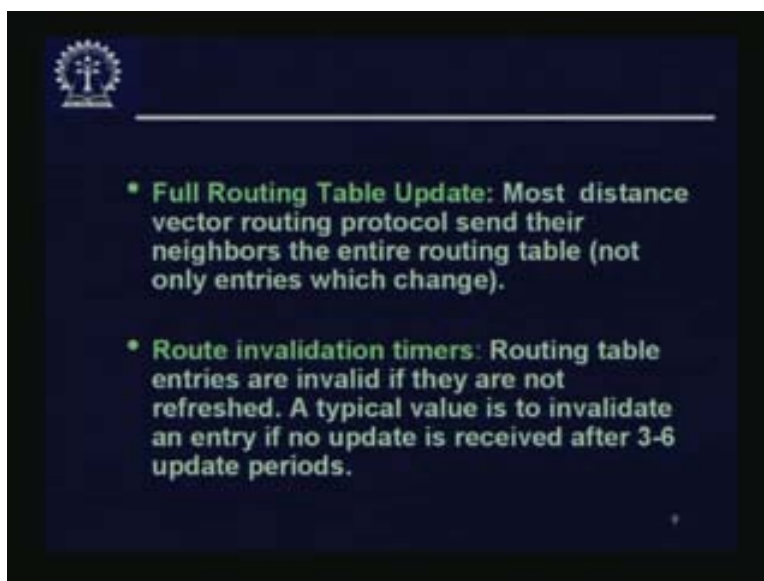
There are two questions here; one is that, what is the routing protocol you are running inside the autonomous system, and what are the routing protocols you are running between the autonomous systems. This is because finally your packet may go to a router to a network to the backbone of the same autonomous system still then it might have to hop across some autonomous systems and then finally enter into another autonomous system and go all the way down to a particular network and then to the intended host. So, autonomous systems introduce another level of hierarchy. Routing is done differently within an autonomous system that is the intra domain routing and between autonomous systems which is the interdomain routing. RIP is suitable for intradomain routing. For the time being we are talking about intradomain routing but will look at interdomain routing later on.

(Refer Slide Time: 13:47 to 14:23 min)



The characteristics of distance vector routing: one thing is it uses periodic updates. Updates of the routing tables are sent at the end of certain time period, a typically value is 90 seconds. This means every 1 and a half minutes you send your updates which are the distance vectors to your neighbors. There may be triggered updates. If a metric changes on a link a router immediately sends out an update without waiting for the end of the outdated period because if some link has gone down or some thing has become really congested or some parameter has changed then at that point a router may not wait for the lapse of that 1 and a half minutes but send immediate updates triggered by some kind of an event.

(Refer Slide Time: 14:55 to 15:35min)

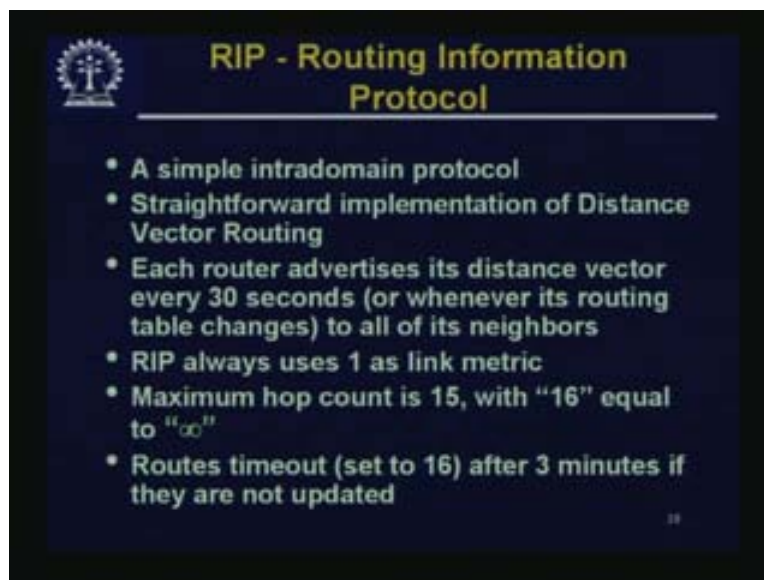


There may also be full routing table updates. Most distance vector routing protocol send their neighbors the entire routing table. Not only entries we change. In some cases of a triggered

update you may send only that data which has changed and then from time to time you send the full which is your distance vector.

Route invalidation timers: Routing table entries are invalid if they are not refreshed. A typical value is to invalidate an entry if no update is received after 3 - 6 update periods. What might happen is that instead of a link going down a router itself may go down and then that router is not sending anymore updates to anybody. But after 3 – 6 update periods its neighbors find that they are not getting any thing then it will set that link to infinity. That route or any route through that particular router which has gone dead will become invalid and this information will slowly percolate through the entire network.

(Refer Slide Time: 16:25 to 18:05 min)



Change is a problem in routing because this change has to traverse globally. Everybody finally have to know but it takes time. We have seen distance vector routing and now we shall look at the actual protocol which is called RIP or Routing Information Protocol. This is a simple intradomain protocol. That means it is in the same autonomous system. Straightforward implementation of distance vector routing is the distributed Bellman Ford algorithm.

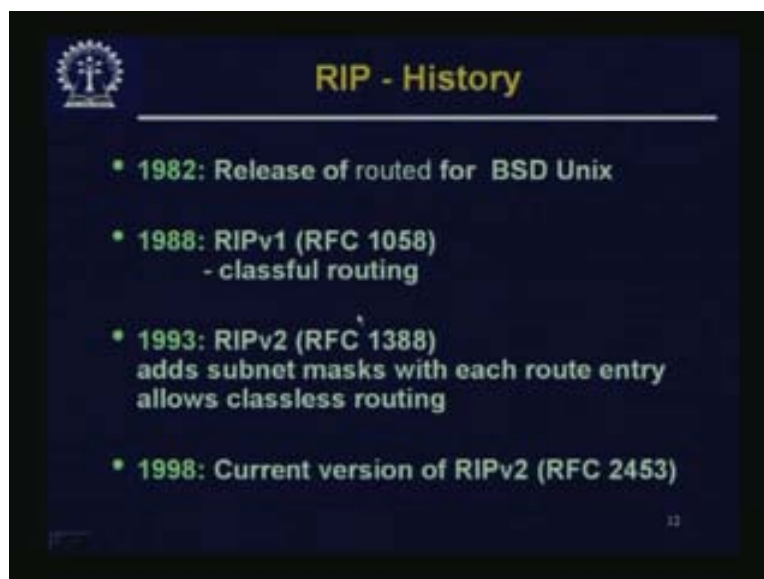
Each router advertises its distance vector every 30 seconds or whenever its routing table changes to all its neighbors. RIP always uses 1 as link metric therefore negative cycles are not present. The maximum hop count is 15 with 16 is equal to infinity which means it is rather a low value for infinity and the reason we need to keep the hop count is that a packet which has been misdirected may have to be dropped. And secondly what we are talking about here is this 15 that means it is giving you the length of the maximum number of hops within this autonomous system, within this domain. It is not that from the source to the ultimate destination it has to be 16 but within the domain there cannot be more than 15 hops. Routes time out is set to 16 after 3 minutes if they are not updated.

(Refer Slide Time: 18:05 to 18:26 min)



A brief history: In the late 1960's distance vectors protocols were used in the Arpanet which is the beginning of this internet era. Mid 1970s: XNS Xerox Network System routing protocol is the precursor of RIP in IP and there were some other versions also.

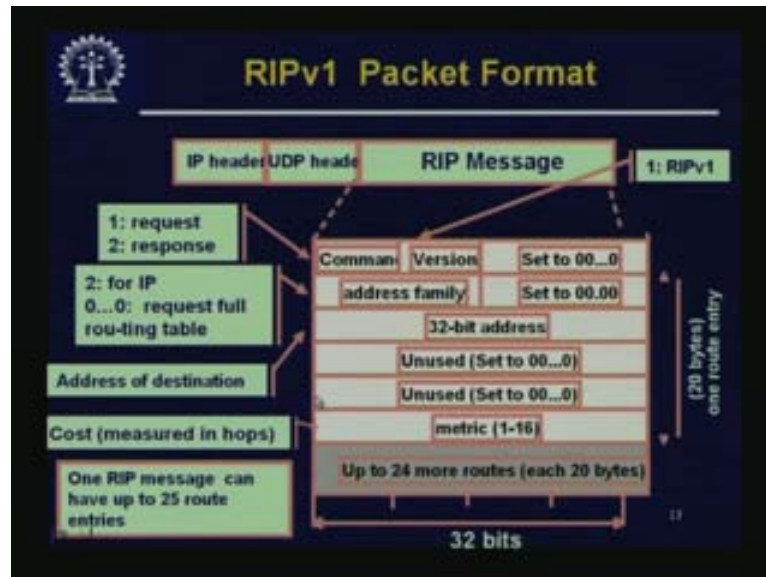
(Refer time slide 18:27 to 19:33 min)



Then this was sort of integrated in BSD distribution of Unix. RIP version one was formalized using this RFC request for comment. This is the standard way of formalizing things in the internet world through RFC's. So, for any topic in this network you can possibly try to find RFC which will give you the most authentic source of information about that. So, version one was described in RFC 1058 and after encountering certain issues with RIP version one people came

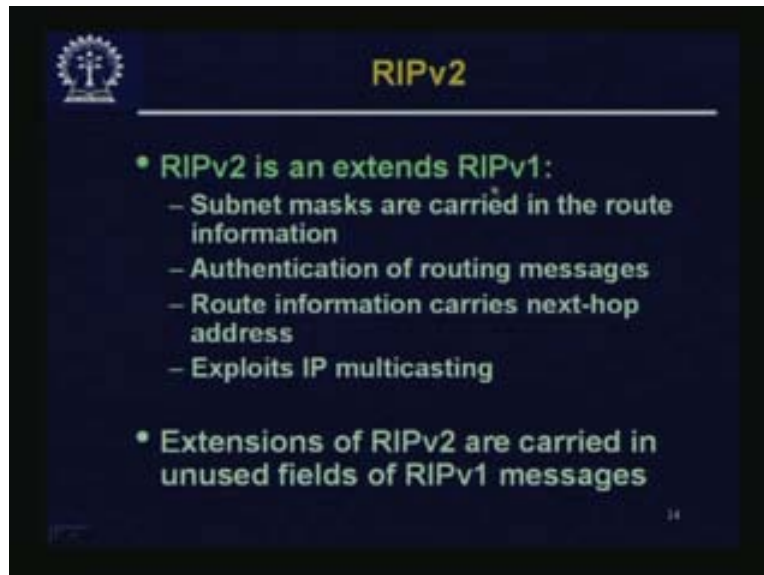
out with RIP version 2 in 1993 that was RFC1388. So it adds subnet masks with IP address with each route entry that allows classless routing. The original version is classful routing then classless routing was also possible with current version of RIP 2 is in RFC 22453 which came out in 98.

(Refer time slide 19:33 to 21:20 min)



What is the packet format of RIPv1? A command would be either request or response. So you may either request some information from some other router or you may be giving information in response to some request. The command would therefore be one or two. And the version would be RIP version one and then you have an address family and this is 2 for Ip, from 0 to 0 this is request full routing table. If you put it this way it means that you are requesting the full routing table. And then you have the address of the destination where you are sending this message and then these parts are unused. So this is 20 bytes, this is one route entry. There may be many such route entries up to 24 more routes each of 20 bytes. So this is actually 32 bits of the total. So, 32 bits means 4 bytes so it is 4 plus 4 is equal to 8, 12, 16, 20 so one RIP message can have up to 25 routing. Then the cost measured in hops is also mentioned.

(Refer Slide Time: 21:21 to 26:27 min)



RIP version 2 is an extension of RIP version one. Subnet masks are carried in the route information. Now what is a subnet mask? In your full IP address which is 4 bytes for the time being in the IP version 4 you have a 4 byte address, so 4 bytes is 32 bits out of which the first part would be the network prefix and the last part is the particular host in the network prefix. Now the question is which part of it or how many bits is the first prefix which shows the network and how many bits are there to show the host? Now there were some standards here these are the so called classful days. But there were class a, class b, class c etc these are classful where it is mentioned that for class a address the first byte would be the network prefix and the last three bytes are for the host which means you can have a very large network with a large number of hosts.

On the other side of the scale you have a class c where 3 bytes would be the network prefix and one byte would be a small network or limited to less than 255 hosts but a larger number of such networks are possible. But even then what happened was that originally no one realized that the network world will grow into such a manner so only 4 bytes were kept as this address and this address space really became small.

Suppose in an organization there are only 30 machines you will network them and then you want to connect them to the internet and for that you have to get some IP addresses. Now, if this network is to be specified completely then it has to be given a unique IP address for the network prefix part. So at best what you can do is that give it a class c address which is say 3 bytes long. So, that 1 byte is left over for up to 255 machines in your organization. But then you are using only 20 of them so a part of the address space is not used.

Now, in order to do that people wanted more flexibility in saying that how many bytes contain the network prefix and how many bytes contain the number of hosts and that would be host part of the address. So you have to specify that how many bits are present. So now we are going from classful routing to classless routing. This classless routing has to carry that information called as

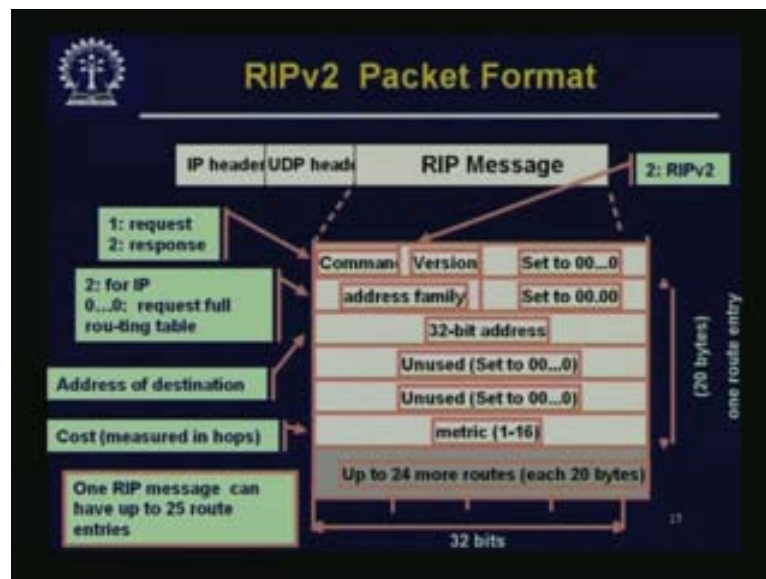
the subnet mask. So subnet masks are carried in the route information so that we can do classless interdomain routing.

Authentication of routing messages was there because routing version one had no security. And if it has no security what could happen is that a router might send all kinds of arbitrary distance vector messages to its neighbors creating problem everywhere. When things become very big then you start thinking of different kinds of possibilities so you try to change so some authentication was included in RIP version 2 although the authentication is not very full proof. Route information carries next hop address and exploits IP multicasting.

What is multicasting?

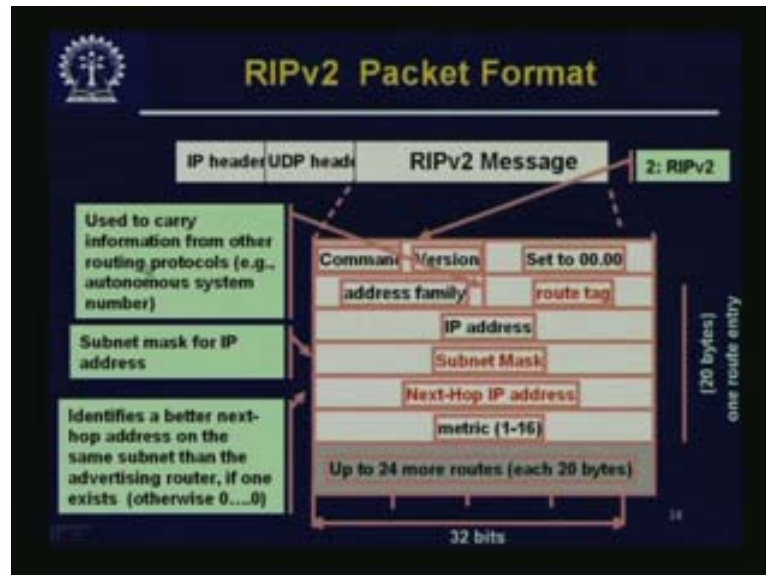
Right now we are talking about unique casting. This means there is one source with one message meant for one destination. Sometimes you want a message to be destined to a group of destinations. You do not want to broadcast it, you do not want to send it to everybody. May be there is a special interest group so you all only want to send to members of this group which is called multicasting. So IP multicasting exploits in some sense. So extensions of RIP version two are carried in unused fields of RIP version one messages.

(Refer Slide Time: 26:27 to 26:55 min)



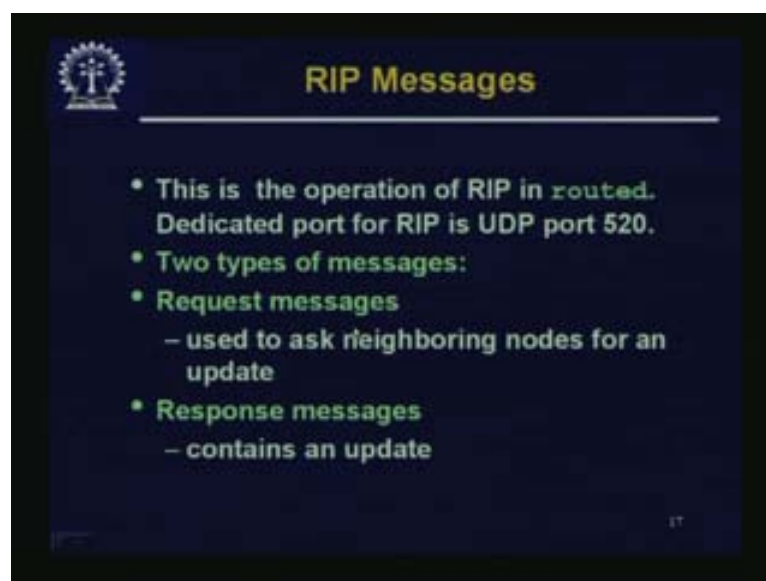
In the RIPv2 packet format as usual the command contains one or two whether it is a request or a response and then in RIPv2 there is the address family, address of destination and then the cost measured in hops.

(Refer time slide 27:11 to 27:56 min)



What happens here is, this field is used for the subnet mask for IP address. So you have an IP address over here, you see the how it is changed and the subnet mask has come here so that now you can do classless routing. Next hop IP address identifies the better next hop address on the same subnet than the advertising router if one exists, so that may come here and these are the changes. And used to carry information from other routing protocols, for example autonomous system, number, etc comes in this part.

(Refer Slide Time: 27:56 to 29:51min)

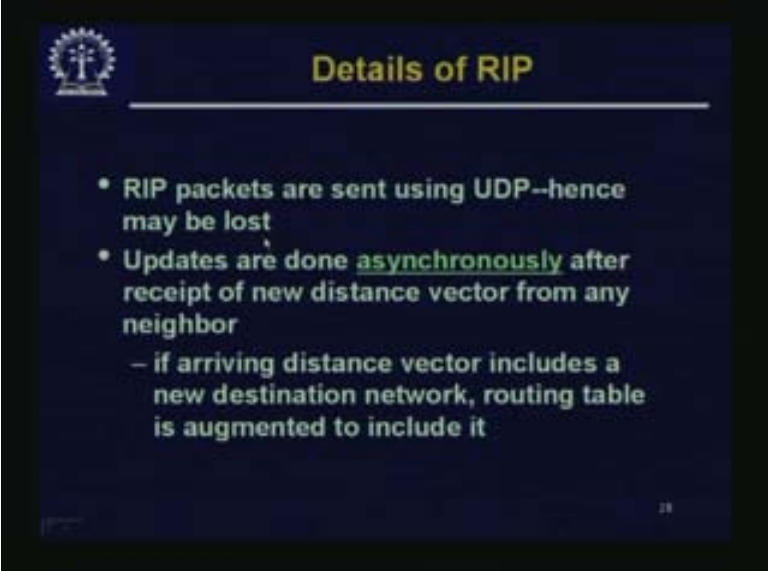


Now a little bit about RIP messages: This is the operation of RIP in routed. So this routed is actually a command. A dedicated port for RIP is UDP port 520. If you remember we had come

across TCP and UDP earlier. These are the transmission level protocols which are at a higher level protocol than these network layers. So UDP is one such which has got very low overhead. So we will talk both about UDP and TCP later on.

But this is a higher level protocol and they use ports. Ports means for any message transferred between two nodes so many different transmissions may be going on from the same node to different processes in different hosts. In order to de-multiplex them, unscramble them, when there is an incoming message and you want to know for which machine it is sent to, then with the help of the IP address you can find out that this particular process is for this particular machine designated by a port number. This is a number which you get from the operating system if you ask for it. So, for RIP there is a particular port which is specified which is 520. For standard protocol there are standard ports which are assigned. There are two types of messages: One is a request message used to ask neighboring nodes for an update and response message which contains an update.

(Refer Slide Time: 29:55 to 03:28min)

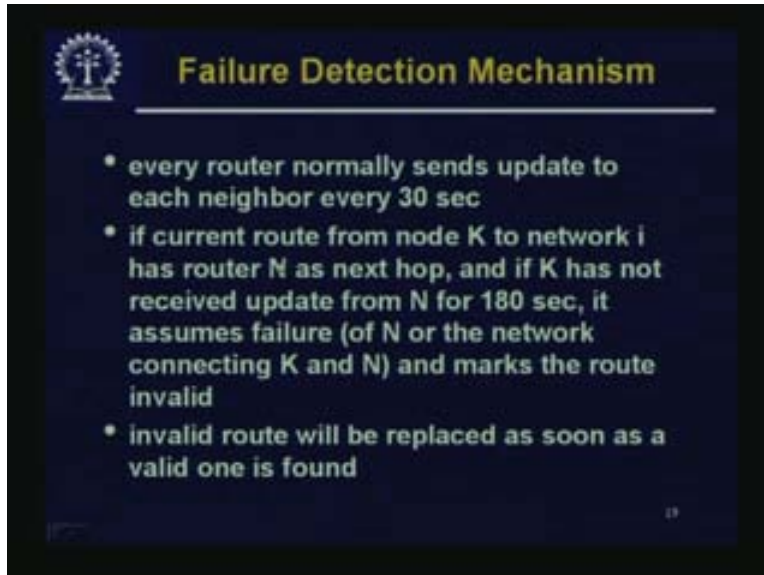


The slide is titled "Details of RIP" in yellow text on a dark blue background. It features a small logo in the top left corner. The content is a bulleted list in white text. The first bullet point states that RIP packets are sent using UDP and hence may be lost. The second bullet point states that updates are done asynchronously after receipt of a new distance vector from any neighbor. A sub-bullet point under the second point states that if an arriving distance vector includes a new destination network, the routing table is augmented to include it. The slide number "28" is visible in the bottom right corner.

- RIP packets are sent using UDP--hence may be lost
- Updates are done asynchronously after receipt of new distance vector from any neighbor
 - if arriving distance vector includes a new destination network, routing table is augmented to include it

RIP packets are sent using UDP hence may be lost. Actually UDP is one transmission protocol which has a low overhead and the other side of the coin is that it is less reliable than its sister TCP which is more reliable so updates may be lost. Updates are done asynchronously after receipt of new distance vector from any neighbor. If arriving distance vector includes a new destination network routing table is augmented to include this. That means routing table now increases in size.

(Refer time slide 30:29 to 31:20 min)



The slide is titled "Failure Detection Mechanism" in yellow text. It features a logo in the top left corner. The content is a bulleted list of three points. The first point states that every router normally sends an update to each neighbor every 30 seconds. The second point explains that if the current route from node K to network I has router N as the next hop, and if K has not received an update from N for 180 seconds, it assumes a failure (of N or the network connecting K and N) and marks the route as invalid. The third point states that an invalid route will be replaced as soon as a valid one is found. The slide number "19" is in the bottom right corner.

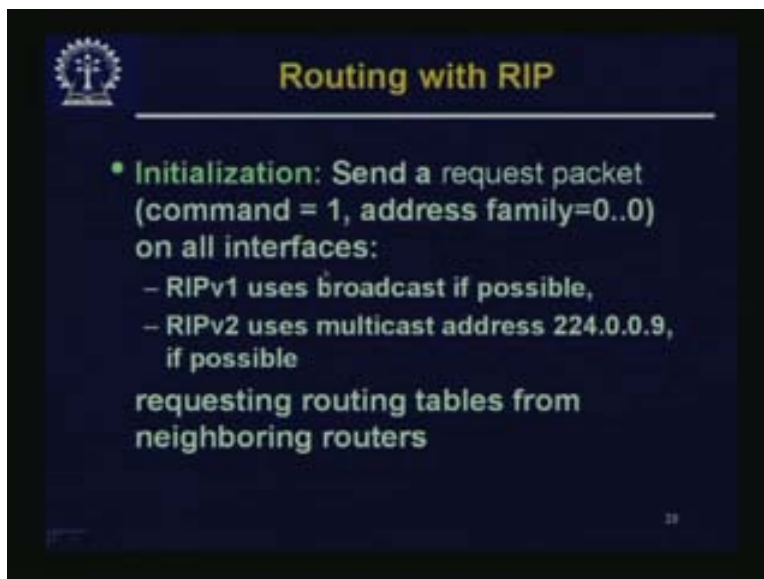
Failure Detection Mechanism

- every router normally sends update to each neighbor every 30 sec
- if current route from node K to network I has router N as next hop, and if K has not received update from N for 180 sec, it assumes failure (of N or the network connecting K and N) and marks the route invalid
- invalid route will be replaced as soon as a valid one is found

19

And that is very important because when you power up your routing table may not contain anything. As you get more and more information from your neighbor your routing table grows. Every router normally sends update to each neighbor every 30 seconds. If the current route from node k to network I has a router n as next hop and if k has not received update from n in for 180 seconds which is 3 minutes it assumes failure. That means 6 such update periods then it assumes failure of n or the network connecting k and n. This does not matter because either ways it is the same for k and it marks the route invalid. Invalid routes should be replaced as soon as a valid one is found.

(Refer Slide Time: 31:20 to 32:20 min)



The slide is titled "Routing with RIP" in yellow text. It features a logo in the top left corner. The content is a bulleted list of two points. The first point is "Initialization: Send a request packet (command = 1, address family=0..0) on all interfaces:". The second point is "requesting routing tables from neighboring routers". The slide number "20" is in the bottom right corner.

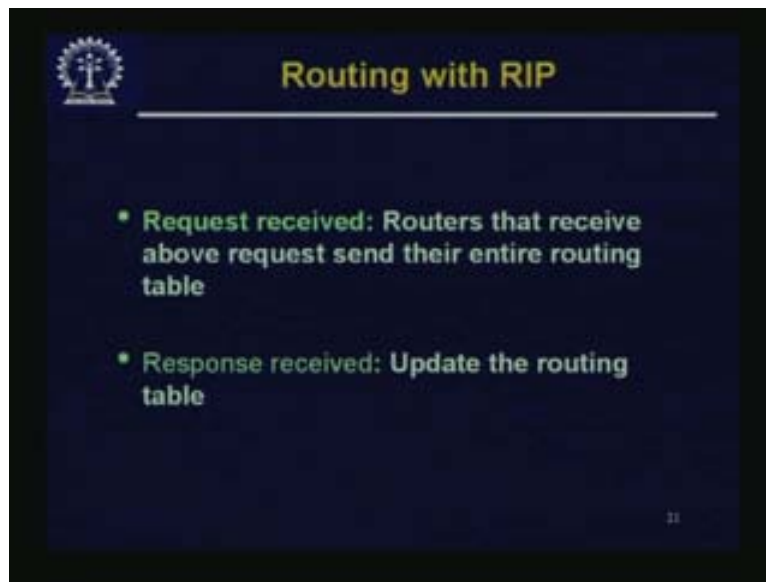
Routing with RIP

- **Initialization:** Send a request packet (command = 1, address family=0..0) on all interfaces:
 - RIPv1 uses broadcast if possible,
 - RIPv2 uses multicast address 224.0.0.9, if possible
- requesting routing tables from neighboring routers

20

How does it initialize? Send a request packet command equal to 1, address family is equal to 0 to 0 that means you are looking for the IP addresses and all interfaces. That means you flood a request at all interfaces. RIPv1 uses broadcast if possible, RIPv2 uses multicast address if possible. The point is that when it is initializing it is asking for information from all possible sources. Of course its sources must come through one of the interfaces so it sends out the request through all the interfaces may be one by one or by some means. Requesting routing tables from neighboring routers means we are trying to get full routing tables from the neighboring routers, that is the initialization phase.

(Refer Slide Time: 32:20 to 32:35 min)



Request received: When routers that receive the above request send their entire routing table and response received update the routing table. Naturally you update the routing table using your distributed Bellman Ford. And then after some time when things are more stable the regular routing updates every 30 seconds send all or part of the routing tables to every neighbor in a response message. And then there are triggered updates. Whenever the metric for a route change something becomes invalid then send the entire routing table.

(Refer Slide Time: 32:59 to 33:12 min)



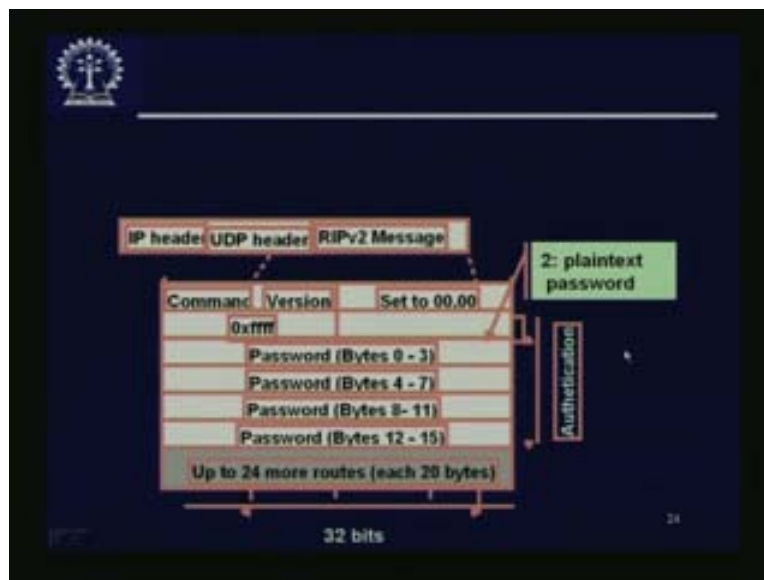
RIP Security

- **Issue:** Sending bogus routing updates to a router
- **RIPv1:** No protection
- **RIPv2:** Simple authentication scheme

23

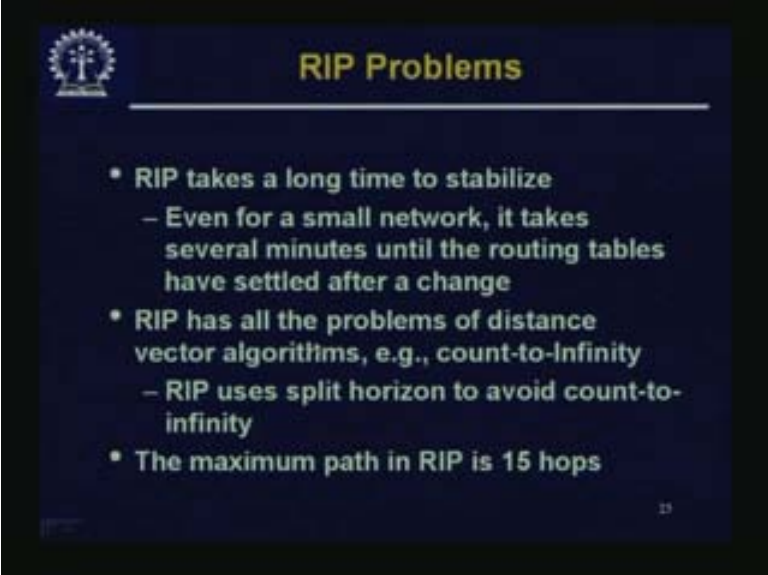
Issue: One is the security issue, sending bogus routing updates to a router. RIPv1 has no protection. RIPv2 uses simple authentication scheme. We have mentioned about this authentication earlier.

(Refer Slide Time: 33:13 to 33:35 min)



What you may have is a few passwords. Unfortunately these passwords are of course plaintext passwords so it does not give you too much of security but at least it is better than RIPv1 which had no security whatever.

(Refer Slide Time: 33:36 to 34:13 min)

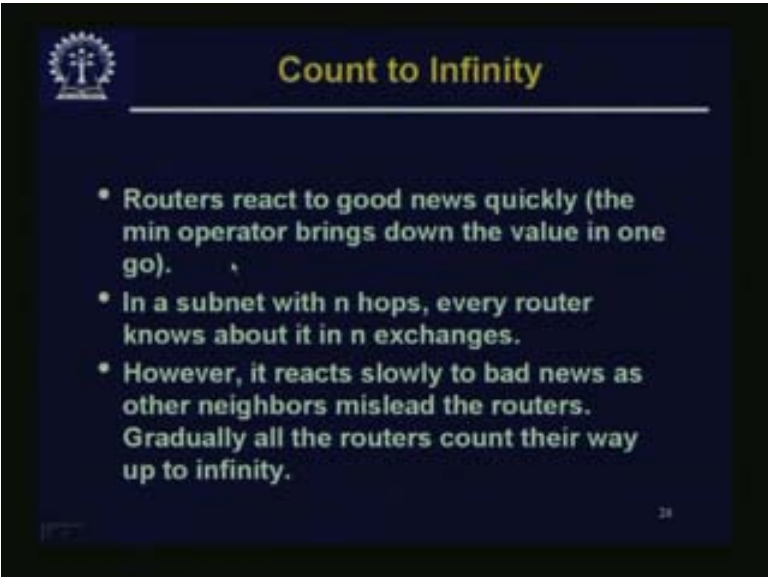


The slide is titled "RIP Problems" in yellow text on a dark blue background. It features a list of three main bullet points, each with a sub-point. The first bullet point is "RIP takes a long time to stabilize", with a sub-point "Even for a small network, it takes several minutes until the routing tables have settled after a change". The second bullet point is "RIP has all the problems of distance vector algorithms, e.g., count-to-infinity", with a sub-point "RIP uses split horizon to avoid count-to-infinity". The third bullet point is "The maximum path in RIP is 15 hops". A small logo is in the top left corner, and the number "25" is in the bottom right corner.

- RIP takes a long time to stabilize
 - Even for a small network, it takes several minutes until the routing tables have settled after a change
- RIP has all the problems of distance vector algorithms, e.g., count-to-infinity
 - RIP uses split horizon to avoid count-to-infinity
- The maximum path in RIP is 15 hops

There are some problems with RIP: RIP takes a long time to stabilize. Even for a small network it takes several minutes until the routing tables are settled after a change. So that is one problem with RIP. And RIP has all the problems of distance vector algorithm, for example, count-to-infinity. We will come to this specific problem later on. RIP uses split horizon that means with reverse poison as it is usually said to avoid count-to-infinity.

(Refer time slide form 34:13 to 34:36 min)

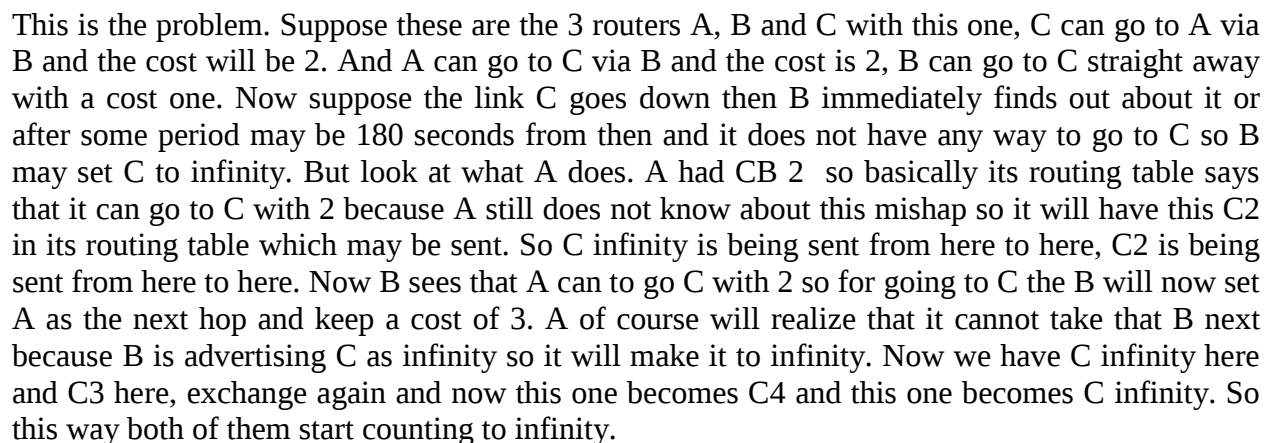


The slide is titled "Count to Infinity" in yellow text on a dark blue background. It features a list of three main bullet points. The first bullet point is "Routers react to good news quickly (the min operator brings down the value in one go)". The second bullet point is "In a subnet with n hops, every router knows about it in n exchanges". The third bullet point is "However, it reacts slowly to bad news as other neighbors mislead the routers. Gradually all the routers count their way up to infinity". A small logo is in the top left corner, and the number "26" is in the bottom right corner.

- Routers react to good news quickly (the min operator brings down the value in one go).
- In a subnet with n hops, every router knows about it in n exchanges.
- However, it reacts slowly to bad news as other neighbors mislead the routers. Gradually all the routers count their way up to infinity.

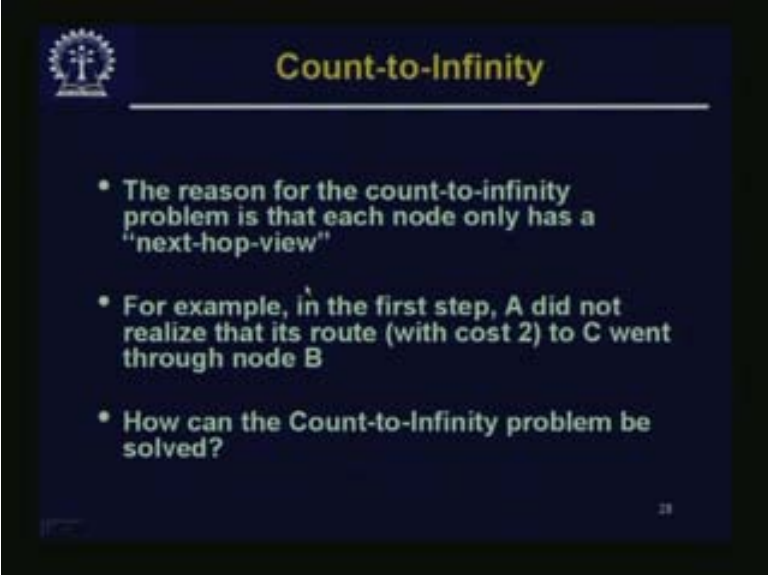
So this is the count-to-infinity problem. The problem is routers react to good news quickly. The min operator brings down the value in one go. In a sub net with n hops every router knows about

(Refer Slide Time: 34:37 to 37:17 min)



What was the problem? The problem was A was sending C to, that means A was going to C via B but it was telling to B that it can go there at a distance 2 without realizing that this might mislead B. So one heuristic which was said is that if A is really going to C through B then to B instead of saying C2 it should say C infinity. So this is the hack so to say which was used.

(Refer Slide Time: 37:18 to 37:38 min)



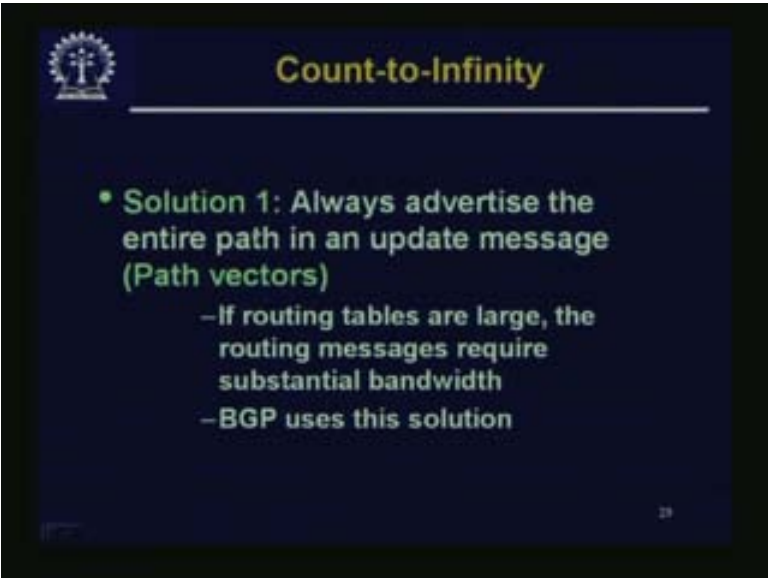
Count-to-Infinity

- The reason for the count-to-infinity problem is that each node only has a "next-hop-view"
- For example, in the first step, A did not realize that its route (with cost 2) to C went through node B
- How can the Count-to-Infinity problem be solved?

28

The reason to the count-to-infinity problem is that each node has only next hop view. For example, in the first step A did not realize that its route with cost 2 to C went through node B. How can count-to-infinity problem be solved?

(Refer Slide Time: 37:38 to 38:12 min)



Count-to-Infinity

- **Solution 1: Always advertise the entire path in an update message (Path vectors)**
 - If routing tables are large, the routing messages require substantial bandwidth
 - BGP uses this solution

29

There is a one solution. This is a proper solution but this is not used in RIP. Always advertise the entire path in an update message. That means you give the entire path. If the entire path was given then this would not have happened. But that is not done. So, if routing tables are large the routing messages require substantial bandwidth. There is an interdomain routing protocol called BGP which uses this solution.

(Refer Slide Time: 38:12 to 38:46 min)

Split Horizon

- A router never advertises the cost of a destination to its neighbor N if N is the next hop to that destination.
- In split horizon with poisonous reverse, it advertises a cost of infinity.
- This does not prevent three way count to infinity.

This split horizon hack is that a router never advertises the cost of a destination to its neighbor n if n is the next hop to that destination. So A never advertises C to B. Or in split horizon with poisonous reverse it advertises a cost of infinity. This does not prevent three way count-to-infinity. But this is a hack this is not a solution because this takes care of some of the cases but this may not take care of all the cases.

(Refer time slide form 38:46 to 40:39 min)

The Reverse-Poison (Split-horizon) Hack

If the path to dest is through neighbor h, report ∞ to neighbor h for dest.

distance tables from neighbors

	A	B	D
A	0	7	∞
B	7	0	∞
C	∞	1	2
D	∞	∞	0

distance tables

	A	B	D
A	1	15	∞
B	8	8	∞
C	∞	9	4
D	∞	∞	2

E's distance table

	A	B	D
A	1	A	
B	8	B	
C	4	D	
D	2	D	

distance table E sends to its neighbors

To	To	To
A	B	D
A: 1	A: 1	A: 1
B: 8	B: 8	B: 8
C: 4	C: 4	C: 4
D: 2	D: 2	D: 2
E: 0	E: 0	E: 0

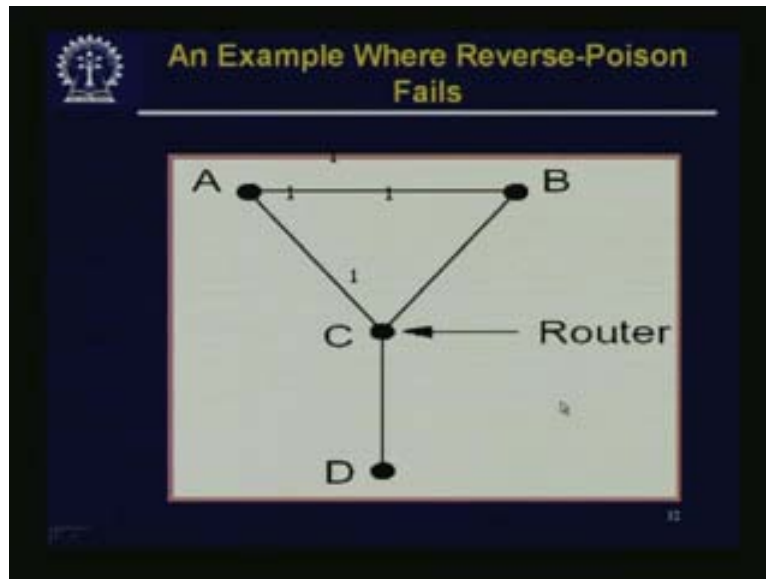
destinations

distance through neighbor

These are the details. Suppose you have this network A B C D E with this kind of cost associated it with it, now consider the router E. So E distance tables from neighbors, from A it may get this (0, 7, infinity, infinity, 1) that means A to A is 0, A to B is 7, A to C is infinity to start with and

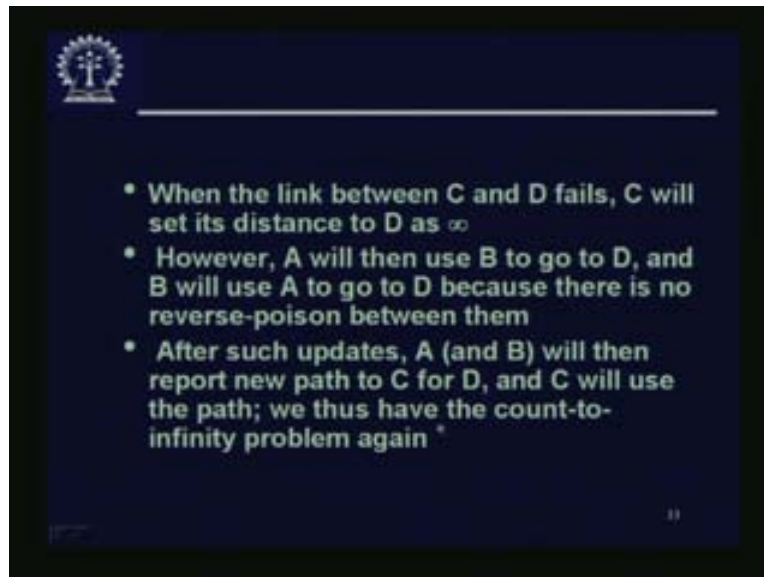
A to B is infinity. So distance table E sends to its neighbors so to A it sends A infinity because it has to go to A itself, actually again it goes directly so to B also it sends infinity, what it sends to D you just look at this value it tells that its cost to C is infinity although to B it says that its cost to C is 4 because to B it says that I can go with a distance 4 because it can take this route from E to C. Whereas it advertises a cost of infinity to D because the path that E thinks it can take with a cost of 4 passes through D so to D it says C infinity rather than C4 which is advertised to A and B. So this is the split horizon with reverse poison.

(Refer Slide Time: 40:40 to 41:05 min)



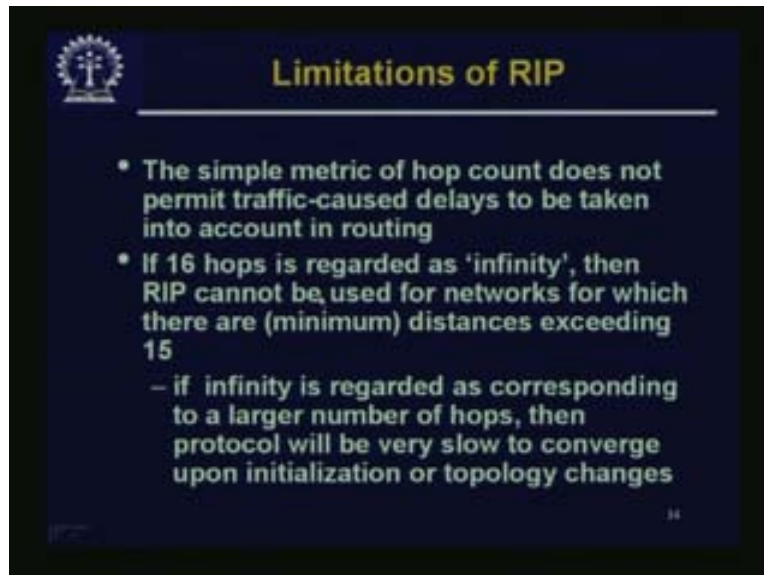
But look at this case. Suppose you have a network structure like this ABC and D. So these are all routers which are very fine. In order to simplify things let us say all the links have a cost of 1.

(Refer Slide Time: 41:05 to 42:42 min)



Now when the link between C and D fails which is this link then C will set its distance to D as infinity. However, A will then use B to go to D and B will use A to go to D because there is no reverse poison between them. What is happening is that previously, B knows that it can go to D through C with a distance 2. A knows that it can go from A to D with a distance 2. Now C is saying D is infinity. But B is saying D is 2. Now B then sends to A and A sends to B and mutually they start counting to infinity not realizing that both of them were actually going through C. But there is no reverse poison between A and B, there is reverse poison between B and C, there is reverse poison between A and C but between A and B there is not reverse poison so they will mutually start counting to infinity. After such updates A and B will then report new path to C for D and C will use the path. Now A will tell C that it has got a path to D through B of course and C will not know and now C will also take that. So for any thing from D it will send it to A, A will send it to B and things like this will go on without any packet getting anywhere. We thus have the count-to-infinity problem again.

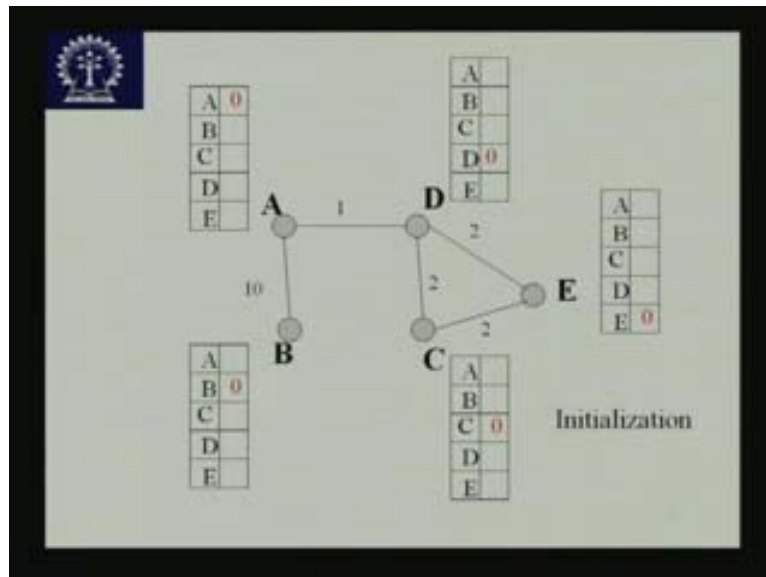
(Refer Slide Time: 42:43 to 44:07 min)



Other limitations of RIP: The simple metric of hop count does not permit traffic-caused delays to be taken into account in routing. So, when we are just talking about the hop count this is a limited kind of metric. So basically we are seeing whether we can reach there at all. But there are other kinds of metrics which may be important. That means some of the links may be very weak or of low capacity then again some links may be very expensive, somewhere the reliability may be poor etc and all these metrics which may be important in today's world where the network communication is very important then you cannot handle it in RIP. So that is one problem limitation of RIP.

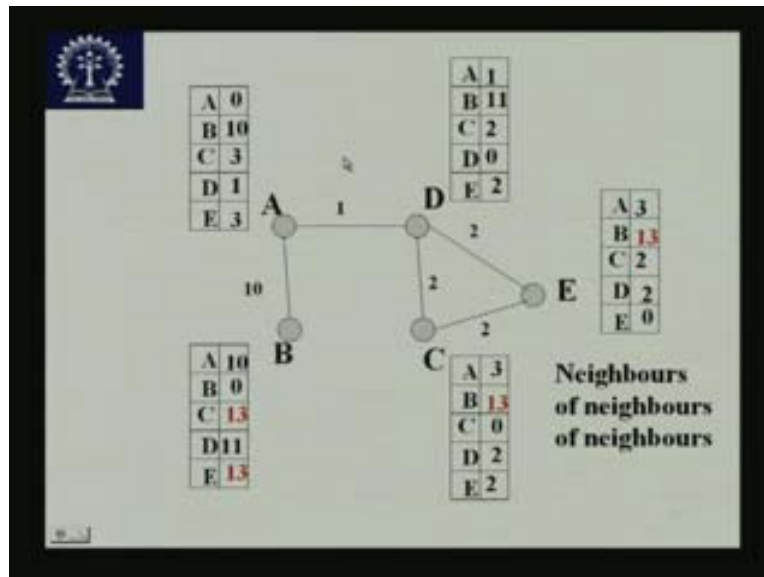
Another limitation is if 16 hops are regarded as infinity then RIP cannot be used for networks for which there are distances exceeding 15. If there are distances exceeding 15 then RIP cannot be used. So, by today's standard that may be a small value. If infinity is regarded as corresponding to a larger number of hops then protocol will be very slow to converge upon initialization or on topology changes. The point is that, if you see a value of 16 or if something has gone up to a value of 16 you know that that is infinity and you can forget about it. Whereas if you simply change that infinity from 16 to 16,000 then you will have to count right up to 16,000 so the thing would be very slow to converge. Convergence rate after topology change or after crash etc becomes an issue in RIP. We will now just look at an example.

(Refer Slide Time: 44:43 to 45:40 min)



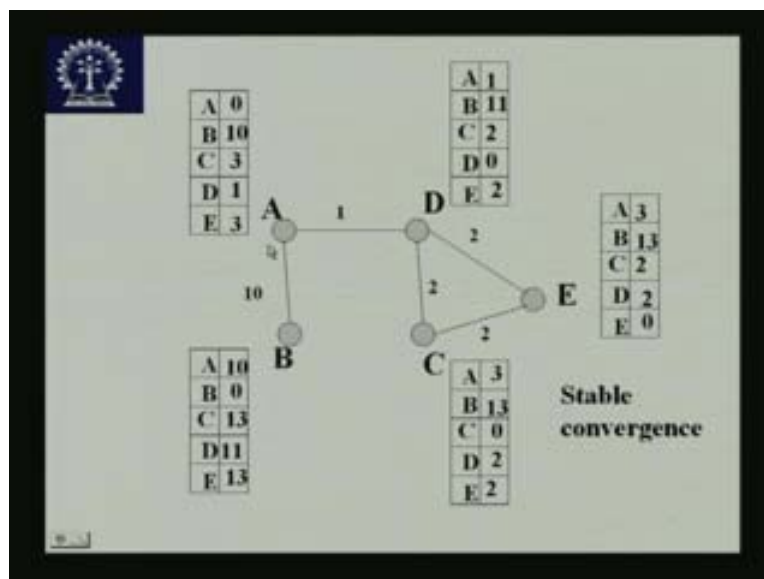
Suppose this is the situation and then in the initialization phase A sets itself to 0, D sets itself to 0, E sets itself to 0 and so on then they solicit router advertisements. Then what will happen is that when they get the distance vector D will respond and B will also respond so A would know that these 2 routers B and D are just neighbors so it can put a distance 1 in B and D. And the same thing goes for all the routers so to get their next step. So, direct neighbors come in after one exchange. If there are values over here like 10, 2, 2 as distances etc these values get transmitted there so we get A0, B10, D1 for A. Similarly for D it gets E2 C2 and A1. Therefore the routing table starts getting filled up this way. Then of course after the second step you get neighbors of neighbors and so on so this routing table has now become 0, 10, 3 etc so this is on going.

(Refer Slide Time: 46:20 to 46:49 min)



Then neighbors of neighbors of neighbors so those are basically 3 hops away and slowly these values are converging to the correct values. From A to D you can go to 1 and A to E you can go with a step of 3, A to C is 3 and A to B is 10 and A to A is 0. So now after three iterations it has stabilized. So this is a stable convergence and no value changes any longer.

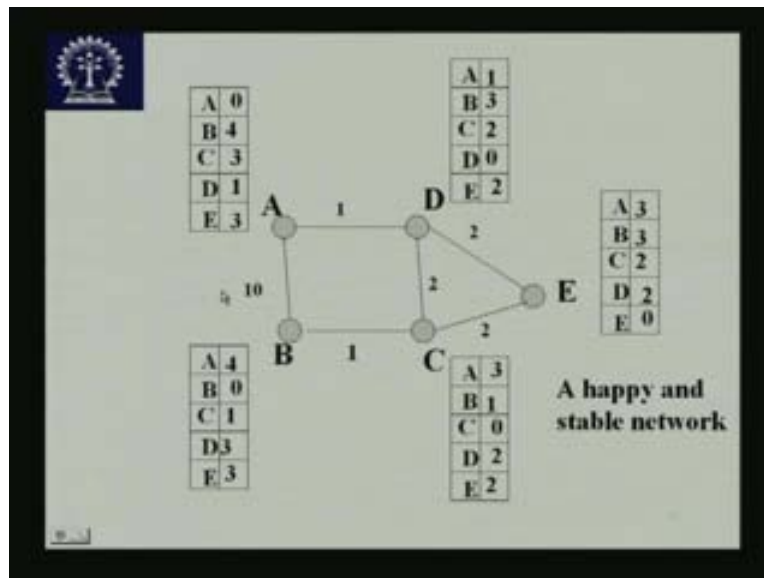
(Refer Slide Time: 46:50 to 48:36 min)



The ones in red were showing the values which change but now there is no change any longer. Now a new link has come up between B and C with a distance of 1. Now this message will start getting transmitted through the network like this. As soon as there is a communication between B and C both of them know that C has B as neighbor with distance 1 and B has C as neighbor with

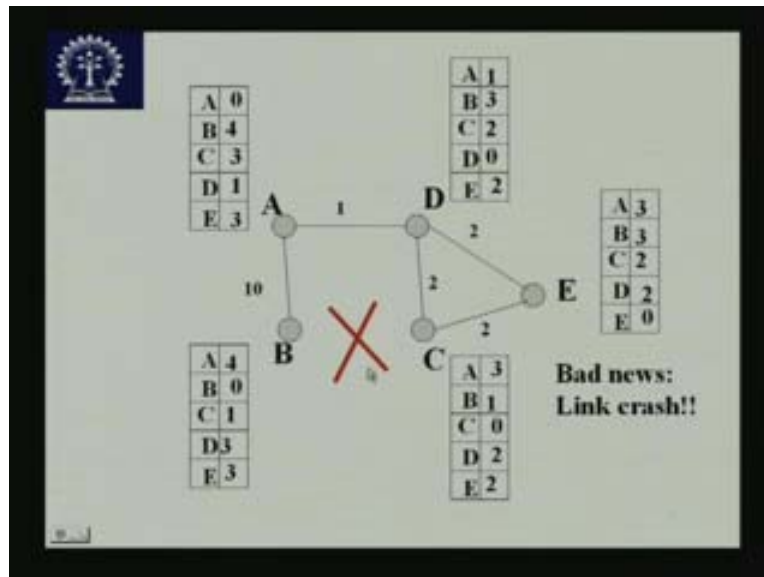
distance 1. So these are the direct endpoints. Then in the next step the neighbor of E will know the neighbor of C which is namely D will also know that they can reach B very quickly. Previously if you remember they had values like 11 and 13 so E had to go to B through 13 when it did not know because it took the route B to D to A to B so that is 2 plus 1 plus 10 is equal to 13. But now since it can go to reach C and 2 and C can reach B with distance 1 so now it has updated the distance to B as 3 and distance to B as 3. So from 11 and 13 these values have come down in 1 hop and A of course still does not know. Now A knows that from going to B it need not take this route which is a very high cost path but it can go via C with a distance of 4. So now neighbors know.

(Refer Slide Time: 48:37 to 48:55 min)



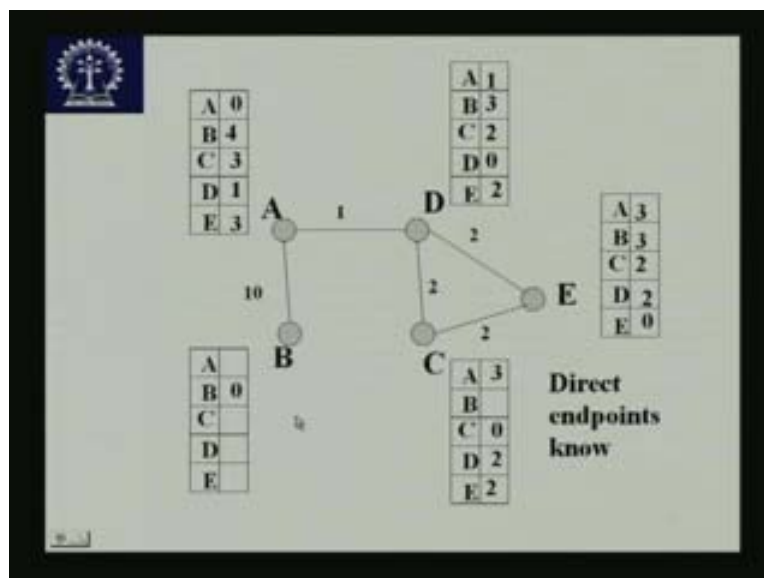
Once again we have a very happy and stable network. The point was that this good news traveled as fast as the number of hops because in one go in the main function will bring down the value to anything.

(Refer Slide Time: 48:59 to 49:08 min)



But now suppose there is a bad news that means this link has crashed then what happens?

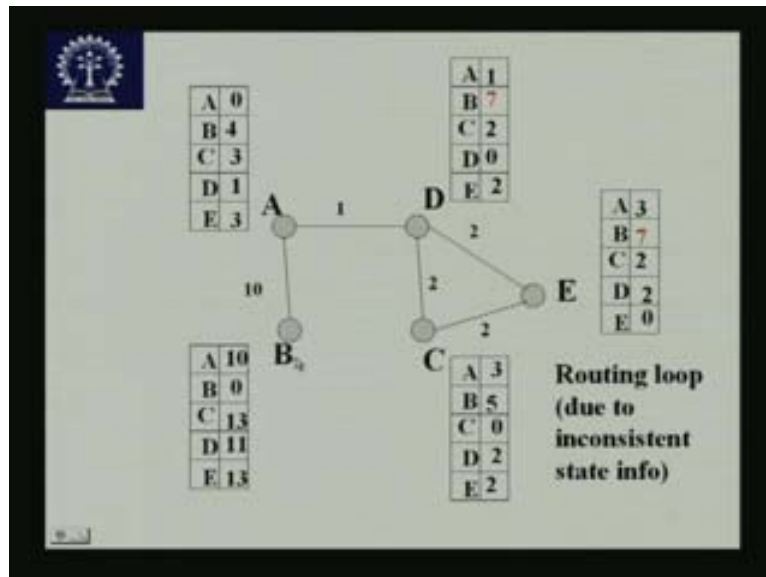
(Refer Slide Time: 49:08 to 49:34 min)



Direct endpoints will know that it can no longer go in between B and C. Previously B for example thought that it can go to A through 4, there should be another entry that it can go through A through C. Similarly it can go to D through C once again, through E to E through C once again. Now when the link from B to C fails all these entries now become invalid because this link has crashed. So, direct endpoints would know that some of their entries will vanish. In the next step it will now set to A as 10 and when it gets the entire thing then it will set to something else, it will get help from neighbors so go to A to 10 and A says that it can go to C

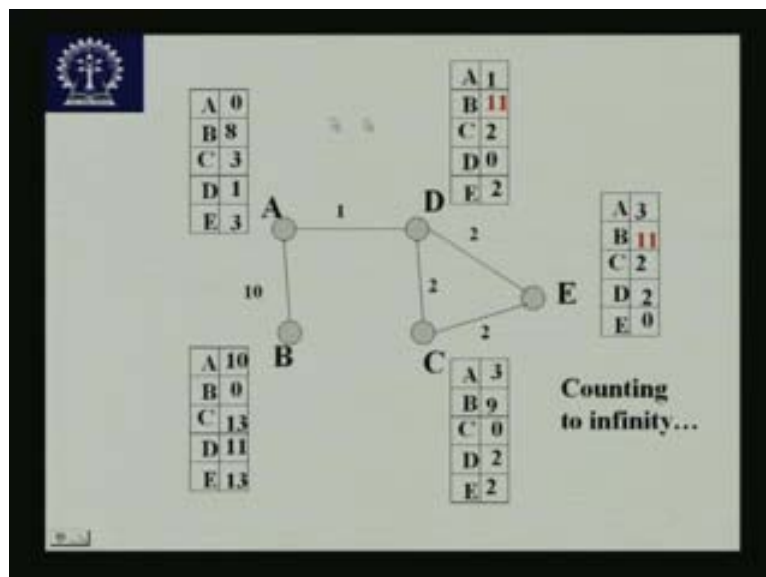
with a distance 3, D with a distance 1 and E with a distance 3 etc so it adds 10 to that so B gets all these values.

(Refer Slide Time: 50:35 to 51:12 min)



Unfortunately there is a routing loop now due to inconsistent state information because now C knows that in order to go to B it can take a path of cost 5, how? It is because there is no way it can reach B through 5. The point is, it has been misled by its neighbors, the same thing that was happening in count-to-infinity.

(Refer Slide Time: 51:15 to 51:37 min)



So they now mutually start counting to infinity for going to B until they reach a value which will force them to go through this link once again. So this count-to-infinity will continue till we will reach the limit and come back to stability once again. These kinds of limitations and problems were there in RIP.

One good thing about RIP was that RIP is a simple scheme, simple algorithm and actually it was conceived in days when networks were smaller but as networks grew bigger and bigger it had problems. this count-to-infinity problem is actually a small problem but other major problems why RIP could not be used when the network really scaled up for very big networks was this problem of having a low value of infinity and making the value larger makes the problem of convergence worse.

And then secondly it is that you are really doing this one domain whereas in today's world with millions of networks you require a hierarchy of domains. You require these autonomous systems and other hierarchy etc. The other limitation you was that we were unable to talk about the quality of the links. Since we were unable to talk about the quality of the links some of the routes which RIP might set up may be much less than optimal.

One thing that is very sure is because there is a lot of difference between the capacity of the links, you get capacities like 64 kbps on one end of the spectrum, you get capacities like 155 megabits per second may be in another side of the spectrum but the point is that these two values are several orders of magnitude away from each other. But RIP has no way of taking all this into account. So this was also important. So what happened was that people moved from this to another algorithm which is sometimes called link state algorithm just as RIP is sometimes called distance vector algorithm because it exchanges distance vectors similarly there is an algorithm which exchanges the state of links globally. Since it exchanges the state of links globally then do some local processing and that algorithm is called Open Shortest Path First OSPF so this has got some advantages over RIP.