WIKIPEDIA

Distance-vector routing protocol

A **distance-vector routing protocol** in data networks determines the best route for data packets based on distance. Distance-vector routing protocols measure the distance by the number of routers a packet has to pass, one router counts as one hop. Some distance-vector protocols also take into account network latency and other factors that influence traffic on a given route. To determine the best route across a network, routers, on which a distance-vector protocol is implemented, exchange information with one another, usually routing tables plus hop counts for destination networks and possibly other traffic information. Distance-vector routing protocols also require that a router informs its neighbours of network topology changes periodically.

Distance-vector routing protocols use the Bellman–Ford algorithm to calculate the best route. Another way of calculating the best route across a network is based on link cost, and is implemented through link-state routing protocols.

The term *distance vector* refers to the fact that the protocol manipulates *vectors* (arrays) of distances to other nodes in the network. The distance vector algorithm was the original ARPANET routing algorithm and was implemented more widely in local area networks with the Routing Information Protocol (RIP).

Contents

Methodology

Routers that use distance-vector protocol determine the distance between themselves and a destination. The best route for Internet Protocol packets that carry data across a data network is measured in terms of the numbers of routers (hops) a packet has to pass to reach its destination network. Additionally some distance-vector protocols take into account other traffic information, such as network latency. To establish the best route, routers regularly exchange information with neighbouring routers, usually their routing table, hop count for a destination network and possibly other traffic related information. Routers that implement distance-vector protocol rely purely on the information provided to them by other routers, and do not assess the network topology.[1]

Distance-vector protocols update the routing tables of routers and determine the route on which a packet will be sent by the *next hop* which is the exit interface of the router and the IP address of the interface of the receiving router. Distance is a measure of the cost to reach a certain node. The least cost route between any two nodes is the route with minimum distance.

Updates are performed periodically in a distance-vector protocol where all or part of a router's routing table is sent to all its neighbours that are configured to use the same distance-vector routing protocol. Once a router has this information it is able to amend its own routing table to reflect the changes and then inform its neighbours of the changes. This process has been described as ‘routing by rumour’ because routers are relying on the information they receive from other routers and cannot determine if the information is actually valid and true. There are a number of features which can be used to help with instability and inaccurate routing information.

Development of distance-vector routing

The oldest routing protocol, and the oldest distance-vector protocol, is version 1 of the Routing Information Protocol (RIPv1). RIPv1 was formally standardised in 1988.[2] It establishes the shortest path across a network purely on the basis of the hops, that is numbers of routers that need to be passed to reach the destination network. RIP is an interior gateway protocol, so it can be used in local area networks (LANs) on interior or border routers. Routers with RIPv1 implementation exchange their routing tables with neighbouring routers by broadcasting a RIPv1 packet every 30 second into all connected networks. RIPv1 is not suitable for large networks as it limits the number of hops to 15. This hop limit was introduced to avoid routing loops, but also means that networks that are connected through more than 15 routers are unreachable.[3]

The distance-vector protocol designed for use in wide area networks (WANs) is the Border Gateway Protocol (BGP). BGP is an exterior gateway protocol and therefore implemented on border and exterior routers on the Internet. It exchanges information between routers through a Transmission Control Protocol (TCP) session. Routers with BGP implementation determine the shortest path across a network based on a range of factors other than hops. BGP can also be configured by administrators so that certain routes are preferred or avoided. BGP is used by internet service providers (ISPs) and telecommunication companies.[4]

Among the distance-vector protocols that have been described as a hybrid, because it uses routing methods associated with link-state routing protocols, is the proprietary Enhanced Interior Gateway Routing Protocol (EIGRP). It was developed by Cisco in the 1980s and was designed to offer better convergence and cause less network traffic between routers than the link-state routing protocol Open Shortest Path First (OSPF).[5]

Another example of a distance-vector routing protocol is Babel.

Count to infinity problem

The Bellman–Ford algorithm does not prevent routing loops from happening and suffers from the **count to infinity problem**. The core of the count-to-infinity problem is that if A tells B that it has a path somewhere, there is no way for B to know if the path has B as a part of it. To see the problem clearly, imagine a subnet connected like A–B–C–D–E–F, and let the metric between the routers be "number of jumps". Now suppose that A is taken offline. In the vector-update-process B notices that the route to A, which was distance 1, is down – B does not receive the vector update from A. The problem is, B also gets an update from C, and C is still not aware of the fact that A is down – so it tells B that A is only two jumps from C (C to B to A). Since B doesn't know that the path from C to A is through itself (B), it updates its table with the new value "B to A = 2 + 1". Later on, B forwards the update to C and due to the fact that A is reachable through B (From C's point of view), C decides to update its table to "C to A = 3 + 1". This slowly propagates through the network until it becomes infinity (in which case the algorithm corrects itself, due to the relaxation property of Bellman-Ford).

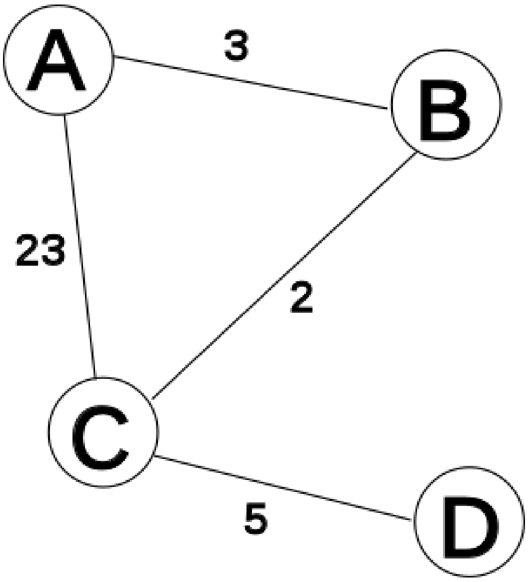
Workarounds and solutions

RIP uses the split horizon with poison reverse technique to reduce the chance of forming loops and uses a maximum number of hops to counter the 'count to infinity' problem. These measures avoid the formation of routing loops in some, but not all, cases.[6] The addition of a *hold time* (refusing route updates for a few minutes after a route retraction) avoids loop formation in virtually all cases, but causes a significant increase in convergence times.

More recently, a number of loop-free distance vector protocols have been developed — notable examples are EIGRP, DSDV and Babel. These avoid loop formation in all cases, but suffer from increased complexity, and their deployment has been slowed down by the success of link state routing protocols such as OSPF.

Example

In this network we have 4 routers A, B, C and D:



We mark the current time (or iteration) in the algorithm with T, and begin (at time 0, or T=0) by creating distance matrices for each router to its immediate neighbours. As we build the routing tables below, the shortest path is highlighted in green, and a new shortest path is highlighted in yellow. Grey columns indicate nodes that are not neighbors of the current node, and are therefore not considered as a valid direction in its table. Red indicates invalid entries in the table since they refer to distances from a node to itself, or via itself.

T=0

from A	via A	via B	via C	via D
to A				
to B		3		
to C			23	
to D				

from B	via A	via B	via C	via D
to A	3			
to B				
to C			2	
to D				

from C	via A	via B	via C	via D
to A	23			
to B		2		
to C				
to D				5

from D	via A	via B	via C	via D
to A				
to B				
to C			5	
to D				

At this point, all the routers (A,B,C,D) have new "shortest-paths" for their DV (the list of distances that are from them to another router via a neighbor). They each broadcast this new DV to all their neighbors: A to B and C, B to C and A, C to A, B, and D, and D to C. As each of these neighbors receives this information, they now recalculate the shortest path using it.

For example: A receives a DV from C that tells A there is a path via C to D, with a distance (or cost) of 5. Since the current "shortest-path" to C is 23, then A knows it has a path to D that costs 23+5=28. As there are no other shorter paths that A knows about, it puts this as its current estimate for the shortest-path from itself (A) to D, via C.

T=1

from A	via A	via B	via C	via D
to A				
to B		3	25	
to C		5	23	
to D			28	

from B	via A	via B	via C	via D
to A	3		25	
to B				
to C	26		2	
to D			7	

from C	via A	via B	via C	via D
to A	23	5		
to B	26	2		
to C				
to D				5

from D	via A	via B	via C	via D
to A			28	
to B			7	
to C			5	
to D				

Again, all the routers have gained in the last iteration (at T=1) new "shortest-paths", so they all broadcast their DVs to their neighbors; This prompts each neighbor to re-calculate their shortest distances again.

For instance: A receives a DV from B that tells A there is a path via B to D, with a distance (or cost) of 7. Since the current "shortest-path" to B is 3, then A knows it has a path to D that costs 7+3=10. This path to D of length 10 (via B) is shorter than the existing "shortest-path" to D of length 28 (via C), so it becomes the new "shortest-path" to D.

T=2

from A	via A	via B	via C	via D
to A				
to B		3	25	
to C		5	23	
to D		10	28	

from B	via A	via B	via C	via D
to A	3		7	
to B				
to C	8		2	
to D	31		7	

from C	via A	via B	via C	via D
to A	23	5		33
to B	26	2		12
to C				
to D	51	9		5

from D	via A	via B	via C	via D
to A			10	
to B			7	
to C			5	
to D				

This time, only routers A and D have new shortest-paths for their DVs. So they broadcast their new DVs to their neighbors: A broadcasts to B and C, and D broadcasts to C. This causes each of the neighbors receiving the new DVs to re-calculate their shortest paths. However, since the information from the DVs doesn't yield any shorter paths than they already have in their routing tables, then there are no changes to the routing tables.

T=3

from A	via A	via B	via C	via D
to A				
to B		3	25	
to C		5	23	
to D		10	28	

from B	via A	via B	via C	via D
to A	3		7	
to B				
to C	8		2	
to D	13		7	

from C	via A	via B	via C	via D
to A	23	5		15
to B	26	2		12
to C				
to D	33	9		5

from D	via A	via B	via C	via D
to A			10	
to B			7	
to C			5	
to D				

None of the routers have any new shortest-paths to broadcast. Therefore, none of the routers *receive* any new information that might change their routing tables. The algorithm comes to a stop.

References

1. Tamara Dean (2009). *Network+ Guide to Networks* (https://archive.org/details/networkguidetone00dean_142). Cengage Learning. pp. 274 (https://archive.org/details/networkguidetone00dean_142/page/n294). ISBN 9781423902454.
2. Hedrick, C. L. "Routing Information Protocol" (https://tools.ietf.org/html/rfc1058.html). *tools.ietf.org*. RFC 1058 (https://tools.ietf.org/html/rfc1058). Retrieved 2019-04-16.
3. Tamara Dean (2009). *Network+ Guide to Networks* (https://archive.org/details/networkguidetone00dean_142). Cengage Learning. pp. 274 (https://archive.org/details/networkguidetone00dean_142/page/n294). ISBN 9781423902454.
4. Tamara Dean (2009). *Network+ Guide to Networks* (https://archive.org/details/networkguidetone00dean_142). Cengage Learning. pp. 274 (https://archive.org/details/networkguidetone00dean_142/page/n294)–275. ISBN 9781423902454.
5. Tamara Dean (2009). *Network+ Guide to Networks* (https://archive.org/details/networkguidetone00dean_142). Cengage Learning. pp. 275 (https://archive.org/details/networkguidetone00dean_142/page/n295). ISBN 9781423902454.
6. RFC 1058 (https://datatracker.ietf.org/doc/html/rfc1058), Section 2.2.2

- "RFC1058 - Routing Information Protocol", C. Hedrick, Internet Engineering Task Force, June 1988 (http://www.ietf.org/rfc/rfc1058.txt)
- "RFC1723 - RIP Version 2 Carrying Additional Information", G. Malkin, Internet Engineering Task Force, November, 1994 (http://www.ietf.org/rfc/rfc1723.txt)
- "RFC2453 - RIP Version 2", G. Malkin, Internet Engineering Task Force, November, 1998 (http://www.ietf.org/rfc/rfc2453.txt)
- "A Path-Finding Algorithm for Loop-Free Routing*, J.J. Garcia-Luna-Aceves and S. Murthy, IEEE/ACM Transactions on Networking, February 1997*
- "Detection of Invalid Routing Announcements in the RIP Protocol", D. Pei, D. Massey, and L. Zhang, IEEE Global Communications Conference (Globecom), December, 2003

Further reading

- Section "Link-State Versus Distance Vector" (http://docwiki.cisco.com/wiki/Routing_Basics#Link-State_Versus_Distance_Vector) in the Chapter "Routing Basics" in the Cisco "Internetworking Technology Handbook"

- Section 5.2 "Routing Algorithms" (http://authors.phptr.com/tanenbaumcn4/) in Chapter "5 THE NETWORK LAYER" from "Computer Networks" 4th. Edition by Andrew S. Tanenbaum

Retrieved from "https://en.wikipedia.org/w/index.php?title=Distance-vector_routing_protocol&oldid=1015206867"