A loop-free path finding strategy to determine the least expensive path is covered in the publication by JJ Garcia-Luna-Aceves and Shree Murthy. The convergence time must be quick and no looping should happen even if the topology changes, which is if a path moves down. The commonly used algorithm, RIP, has bouncing issues and counts infinitely. Even though this is fixed in OSPF, EIGRP, and BGP, all of them still experience brief loops, hence this work offers a loop-free pathfinding technique (LPA). The Basic Distance Vector method, which lists each node's destination, cost, and subsequent hop, is based on the Distributed Bellman Ford Algorithm. The packet's destination is the location it needs to go, the cost is the time it will take to get there, and the next hop is the node it needs to connect to after that. This technique follows the practice of periodically transmitting updates to the routing table to each connected node. The table is updated when a neighbor sends a message. The routing table will be updated following the routes determined by this method by comparing which route's cost is the lowest the present one or the one via a neighbor. When a link changes, if its cost drops from what it was previously, the change is quickly reflected because the value has just decreased, but if its cost rises, it takes much longer to reflect because, for example, if your previous cost was 5 and it has now increased to 22, you will discover this after 17 hops. Because it will continue to bounce between these two nodes 17 times, it is known as the "bouncing effect." The aforementioned phenomenon also results in a count to infinity because if the sole links to the destination break, the packet will keep bouncing between the first two nodes, forming a cycle. We require an LPA because we want to prevent these occurrences. It combines poison reverse and split conversion; in split conversion, a node that has a direct path to another node waits for the neighbor to react with its cost before deciding on the minimal path. Poison reverse is used to get rid of two node loops since the node will inform its neighboring node that the cost to go there is endless even when it isn't. If nothing happens for 90 seconds and the hop count reaches 16, indicating that the packet is bouncing between two nodes, RIP sets the hop count to a maximum of 16, at which point the packet is marked as timed out. This helps to solve the issue we previously addressed. While updates are delivered every 5 seconds, modifications to the routes are sent instantly. Even in the absence of changes, an update is provided every 30 seconds. Even transient loops brought on by the aforementioned techniques will be eliminated by LPA. Each router in LPA keeps track of three tables: The routing table for the destination, next hop, and previous hop in order to get the whole path, the link cost table with the links and their corresponding costs, and the distance table to acquire the destination, neighbor, cost, and predecessors. Because we discover the broken connection while transmitting packets, we may go back to the predecessor because we have the knowledge and choose a different path thanks to this table. The full path may be calculated using prior nodes, and cycles and short-term loops can be validated. The most recent information must be updated on all pathways, and if a connection fails, the failure must be represented throughout the whole table. To avoid it during packet transmission, the path that contains the broken connection must also be changed. Temporary loops are a major problem, so LPA tries to find them before moving on to the next node. If it does, it uses the feasible distance, which compares whether the cost of the target node is lower than that of all others and whether its feasible distance is lower than the current feasible distance. If this is the case, move on to the next node or speak with your neighbors. When the routing mode is passive, there is no looping threat, and when it is active, there is a looping threat, so start chatting with your neighbors to find the next best practical distance. The LPA algorithms employ predecessors as well, offer alternate approaches, and use feasible distance to identify temporary loops, but the BFA methods have possible risks for the count to infinity or bouncing.