Maintenance Phase

- The maintenance phase begins when some node loses its last route due to an adjacent link failure.

- In other words, a node loses its last outgoing (downstream) link.

- We consider that a node has a route to the destination as long as there is at least one downstream link from the node.
Route Maintenance by a Node

- Assume that node i has lost its last downstream link.
- If i does not have any upstream neighbour, it means that there is no other node routing through it.
- If i needs a route for its own use, it discovers one through a QRY-RPY mechanism as in the initialization phase.
Node has Upstream Neighbours

- If the node $i$ has at least one upstream neighbour, it starts a route maintenance phase.
- Node $i$ broadcasts a failure-query (FQ) message regardless of whether it requires a route to the destination.
- The FQ message erases invalid routes. A route is called invalid if it is not rooted at the destination.
Purpose of the FQ Message

- An **FQ message** from i has two different roles:
  - To tell any **upstream neighbours** not to route through i
  - To ask the **upstream neighbours** if they have any alternate routes

- The similarity between **QRY** and **FQ** is that both of them ask for alternate routes to the destination.

- However, **FQ** erases routes simultaneously.
Neighbour’s Action

- When an upstream neighbour (say, j) receives the FQ, several actions may be taken by the neighbour:
  - First, the link between i and j becomes undirected or unassigned.
  - The neighbour j determines whether it still has a route to the destination. It sends a RPY message back to i if it has a route. In that case, j becomes a downstream neighbour of i.
  - If j does not have a route, it propagates the FQ message to its upstream neighbours.
Upstream FQ Propagation

- Once an FQ is transmitted, the upstream FQ propagation continues:
  - It erases invalid routes
  - It propagates until a node is found that has an alternate route. Such a node generates a RPY message.
  - Otherwise, the FQ propagation halts when it erases all invalid routes and cannot find a node with an alternate route. The network is partitioned in this case.
An Example of FQ-RPY Propagation

Nodes that have received FQ
Nodes that have received RPY

Link failure
An Example of FQ-RPY Propagation

Link failure

Nodes that have received **FQ**

Nodes that have received **RPY**
An Example of FQ-RPY Propagation

- Nodes that have received FQ
- Nodes that have received RPY

Link failure
An Example of FQ-RPY Propagation

- Nodes that have received **FQ**
- Nodes that have received **RPY**

Link failure
An Example of FQ-RPY Propagation

Link failure

- Nodes that have received **FQ**
- Nodes that have received **RPY**
Loop Freedom

- The protocol builds and maintains loop-free routes in a static topology.
- However, in a dynamic topology, a downstream blocking rule must be added to ensure loop-freedom:
  - When a node receives a RPY over an unassigned link, it checks whether it has any upstream links. If it does not have any upstream link, it marks the link (over which the RPY was received) as downstream; otherwise, the link is marked downstream-blocked (DN-B).
Downstream Blocking Rule
**Downstream Blocking Rule**

- A node with a **DN-B** link cannot use that link as a downstream link for the purpose of routing.

- To see why this prevents loops:
  - Node i may have a route that passes through j and hence adding a downstream link from j to i may form a loop.
  - However, if j has no *upstream neighbours*, there is no possibility of a route from i through j.

- A **DN-B** link can be converted into a downstream link when j’s upstream links become unassigned.
A DN-B link becomes a Downstream Link

- Since the link i-j was already unassigned, j must have received an FQ earlier.
- Once j propagates this FQ to its upstream neighbours, all upstream links will be unassigned.
- The j-i link can be made downstream then.
Deadlock Freedom and the FQ Wait Rule

- We need to discuss the possibility of a deadlock due to the FQ Wait Rule:

- A node that transmits a FQ and has upstream neighbours, will wait and not transmit another control packet until it has received a transmission from each of those neighbours.

- The purpose of this rule is to erase the invalid routes first before establishing new routes.
The Implementation of the FQ Wait Rule

- This rule is implemented through link markers. When a downstream node transmits an FQ, it temporarily marks each of its upstream links as unassigned waiting (UN-W).
- Similarly, whenever an upstream node receives a FQ over a downstream link, the link is temporarily marked as awaiting broadcast (A-BR).
UN-W and A-BR Links

UN-W

FQ

A-BR

FQ

UN-W

FQ

UN-W
Deadlock Cannot Occur

- The presence of a UN-W marker places a node in a **wait state** prevents it from transmitting until it has received either a RPY or a FQ transmission over each of the UN-W links.
- Similarly, the presence of an A-BR marker forces a node to eventually broadcast either a RPY or a FQ.
- It can be shown that it is not possible to form a cycle of waiting nodes and hence deadlock cannot occur.
Potentially Unbounded Instability

Link failure

- Nodes that have received **FQ**
- Nodes that have received **RPY**
Potentially Unbounded Instability

If j transmits before i, i will drop the RPY

Nodes that have received FQ
Nodes that have received RPY

Link failure
Potentially Unbounded Instability

Nodes that have received FQ

Nodes that have received RPY
Potentially Unbounded Instability

The network settles down fast.

Nodes that have received FQ
Nodes that have received RPY

Link failure
Potentially Unbounded Instability

If $i$ transmits before $j$, a chase starts.

- Nodes that have received **FQ**
- Nodes that have received **RPY**
Potentially Unbounded Instability

If node $i$ transmits before node $j$, a chase starts.

- Nodes that have received FQ
- Nodes that have received RPY

Diagram:
- Link failure
- Node $i$ transmits before node $j$
Potentially Unbounded Instability

If \( i \) transmits before \( j \), a chase starts with FQ chasing RPY.

- Nodes that have received FQ
- Nodes that have received RPY

\( D \)
FQ Chasing RPY

- The chase stops when FQ catches up with RPY.

- Any RPY message in a partition of the network without the destination is inherently false and it is called a false reply.

- It has been shown that under random scheduling of packet transmission, the network settles down fast by erasing all invalid routes.
Properties of the LMR Algorithm

- The main drawback of the LMR algorithm is the potential for transient periods of instability.
- Though it may occur rarely in practice, there is no mechanism in the protocol itself to prevent such instabilities.
- Otherwise, the LMR protocol has many good features such that low control messages, any change affecting only local areas and low latency in route finding.