IV. DUAL REINFORCEMENT Q-ROUTING ALGORITHM

Backward exploration together with the forward exploration is applied in DRQR algorithm in order to improve the learning rate of the Q-Routing algorithm. In DRQR, exact delay values learnt from the backward learning have also been used in the routing tables in addition to the estimation values learnt from forward learning in Q-Routing. This has doubled the learning information available in the algorithm thus improved the learning rate of the algorithm.

In DRQR algorithm, when *x* sends a packet to node y to get its estimated remaining trip times, y also gets *x*'s estimated trip times for its link with *s*.

$$Q_x(z',s) = \min_{z \varepsilon N(y)} Q_y(z,s) \qquad (3)$$

$$\Delta Q_y(y,s) = \eta_b (Q_x(z',s) + q_y - Q_y(x,s)) \qquad (4)$$

In fig. 2, packet at node *x* arriving from source node s is sent to node y, also carries the estimated time that it takes from node *x* to *s*, $Q_x(\acute{z},s)$ (equation 3). With this information node *y* updates its own estimate $Q_y(x,s)$ for the entry node *x* associated with the destination *s* (equation 4). Therefore, in DRQR both backward and forward exploration can be used to update the Q entries. However, this adds an overhead to the packet and to the algorithm.

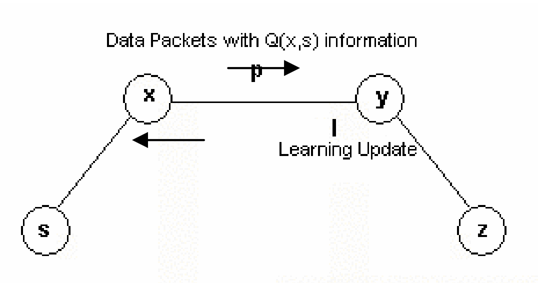

Figure 2: Forward and backward exploration.

In fig. 2, during the forward exploration, the node sending the data packet (node x) updates its $Q_x(y,d)$ value pertaining to the remaining path of packet *p* via node *y*. However, during the backward exploration, the receiving node *y* updates its $Q_y(x,s)$ value pertaining to the traversed path of packet *p* via node *x*. Main drawback of the both Q-Routing and DRQR approaches is that, although they are capable of detecting the link and node failures, they are incapable of restoring them.

V. IMPLEMENTATION AND EXPERIMENTS

SPR, Q-Routing and DRQR algorithms were implemented in the following environment:

- All of the algorithms are implemented in the C language in a parallel environment by using PVM under Linux OS.
- Assigning every node to a different process both on the same and distributed machines simulates parallel behaviour.
- Random uniform traffic distribution is used with varying packet creation rates.
- NSFNET (fig. 3) is used as the network topology with each link having equal cost.
- For each simulation, packet generation is stopped after creating 5000 packets per node and simulation is stopped after all packets are arrived to their destinations or detected and deleted from the network.
- Every simulation is run 3 times and the average of the results is used for accuracy.
- A random link fails every 5 seconds for a period of 4 seconds and traffic is directed to other neighbours.
- It is assumed that there is no packet loss.
- Packet payload is fixed (512 Kbytes).
- In OSPF, all the nodes are assumed to be in the same area (area 0) or no hierarchal routing is considered.
- The HELLO packets are exchanged between the nodes every 5 seconds to know whether its neighbouring nodes are active.
- The dead interval taken as 4 times the HELLO interval, which is the time duration the node decides if its neighbouring node is dead if doesn't get a HELLO packet from it [5].

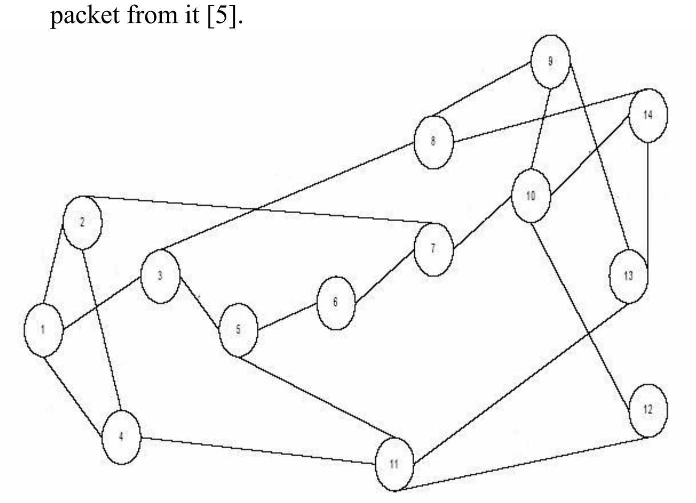

Figure 3: NFSNET topology.