Because Of it's low diameter and latency dragonfly model plays an important role in exascale architecture. In this paper, the prime focus is on the modeling and simulation of large-scale dragonfly networks using the ROSS. The paper demonstrates the performance of ROSS on both the Blue Gene/P and Blue Gene/Q systems on a dragonfly model with up to 50 million nodes. The dragonfly network model for million-node configurations strongly scales when going from 1,024 to 65,536 MPI tasks on IBM Blue Gene/P and IBM Blue Gene/Q systems. With ROSS, the dragonfly network model attains a peak event rate of 1.3 billion events per second.

What is Blue Gene???: **Blue Gene** is an IBM project aimed at designing supercomputers that can reach operating speeds in the petaFlops(10^15 floating point ops) range, with low power consumption.

Simulation Of the DragonFly model:

1.One way to support a low latency and low diameter interconnection network is to efficiently utilize high-radix routers(many narrow ports generating pin bandwidth creating low latency and cost).

a) Topology: It uses a hierarchical topology consisting of nodes connecting to each other(local and global) forming groups and in every group there is a router. So the degree or radix has a formula for load balancing(less equal traffic distributed across nodes) the traffic network.
b) Traffic and Buffering: The model follows two traffic patterns.
   Uniform Random Traffic: With uniform random traffic, each packet generated by the model randomly chooses a destination node from a uniform distribution.
   Nearest Random Traffic: Dragonfly model was tested on small numbers of nodes i.e. Booksim model.In this model dragonfly uses this traffic pattern.

They applied VCs(virtual channels) to avoid deadlock in routing means several VCs maintain a finite buffer and connect to the physical Channel.

DragonFly routing Algorithms: It follows two algorithms a)minimal routing algorithms b) non minimal adaptive algorithms
Under uniform traffic min routing works smoothly but under neighbour traffic pattern it fails as it always uses a single global channel to route hence congestion will increase so we use the other algorithm, the adaptive one.

DragonFly discrete event simulation:
In the dragonfly model, each LP(logical processes) represents an individual router or node and each time-stamped message represents a packet sent to/from a node/router.

Algorithm 1 **packet generate event**

1:if traffic==nearest neighbor then
2: select random destination node ID from the neighboring group
3: else
4: select a random destination node ID from any group
5: end if

6: record packet destination and travel start time in the packet header
7: schedule a packet send event on current node
8: schedule the next packet generate event after time delay tD (exponential distribution).

Algorithm 2 **packet send event**

1: check the buffer count of the downstream virtual channels
2: if buffer count < BUFFER SIZE then
3: schedule a packet arrive event on the source router at output port available time
4: update output port available time
5: buffer count++ 6: else
7: keep rescheduling the packet send event until virtual channel buffer space is available
8: end if

Algorithm 3 **router lp arrive event**
1: assign an input port to the packet
2: find an available input virtual channel for the packet
3: if input virtual channel is available then
4: schedule the packet on corresponding input port at input port available time
5: update input port available time
6: schedule a router send event
7: keep holding input virtual channel until an output channel is available
8: else
9: keep rescheduling event until input virtual channel is available at the input port
10: end if

Algorithm 4 **router send event**
1: assign an output port to the packet
2: check the buffer count for the downstream output virtual channels
3: if buffer count < BUFFER SIZE then
4: schedule the packet on output port at output port available time
5: release the input virtual channel
6: update output port available time
7: send credit message back to packet sender node/router LP
8: if packet destination node is connected to router then
9: schedule a packet arrive event on node
10: else
11: schedule a router arrive event on router
12: end if
13: else
14: keep rescheduling the router send event until output virtual channel is available
15: end if

Algorithm 5 **packet arrive event**
1: if node id == packet destination node then
2: collect statistics (end-to-end latency, number of packets completed)
3: else

4: display error
5: end if

Validation occurs of model's behavior and performance by comparing with the serial cycle accurate interconnection network simulator, booksim, that was used to validate the dragonfly network topology proposal .They verified that the dragonfly network model agrees with the booksim simulator under a variety of packet arrival rates, routing algorithms, and traffic patterns.