ABSTRACT

Rapid growth of Internet traffic has thrown up a new challenge in the design of packet processing and switching functionality. Need arises for flexible, wire speed packet processing devices, which can be rapidly adapted to the continuously changing standards and end-user requirements. High speed switches are the key components of the interconnection networks used in multiprocessors and integrated communication networks for data, voice, and video. Packet switches with a crossbar switch fabric and virtual output queues at the inputs are attractive because of their implementation feasibility and low memory bandwidth requirement.

This thesis involves comprehensive study, simulation and implementation of packet processing and packet switching architectural aspects.

In the initial part of this thesis, we summarize our comprehensive study of different generation of network system architectures including network processor and switching fabric architecture.

In subsequent part, the packet processing functions of network processor i.e (1) Packet encryption (International Data Encryption Algorithm as a core of Network Processor) and (2) Packet classification (by three different existing methods) have been implemented with an objective of evaluation and demands on VLSI area.

We also study, simulate, implement and compare space division packet switch configurations like Knockout switch, Batcher Banyan with Trap and Crossbar switch fabric with various scheduling algorithms for subsequent comparison with our proposed scheduling algorithms.

Major contribution of the thesis is to propose two algorithms, m-DPA and DSA. Our Modified Diagonal Propagation Arbiter (m-DPA) scheduling algorithm saves loss of the cells due to buffer overflow, by rotating the priority based on queue occupancy. Dynamic Scheduling Algorithm (DSA), proposed by us, incorporates Quality of Service (QoS) requirement and rotates the priority based on queue occupancy, and prioritized QoS. Performance parameters obtained with our proposed scheduling algorithms are compared with those obtained with existing scheduling algorithms. The results indicate improved throughput (efficiency), operating frequency, and QoS support in DSA.