Design and Implementation of High Performance Transfer Protocol UDT

Akshat Marya[1], Sumit Kumari[2]

[1]4th year CSE, Echelon institute of Technology, Faridabad, India

[2]Assistant Professor, Gurgaon Institute of Technology and Management, Gurgaon, India

Abstract: ***Efficient and reliable communication is essential for achieving high performance in a networked computing environment. Limited network resources bring about unavoidable competition among in-flight packets, resulting in network congestion and, possibly, deadlock. The effectiveness of deadlock handling mechanisms in a network often determines the upper bound in the throughput and utilization that the network can deliver. The proposed mechanism is based on increasing the scheduling priority of packets involved in congestion and providing necessary resources for those packets to make forward progress. Simulation results show that the proposed technique not only disperses network congestion effectively but also contributes to achieving high network performance.***

Keywords: Interconnection network, deadlock-free routing, message dependency, parallel processing.

1. Introduction

The exploitation of parallelism in solving large scale problems has given rise to the parallel computing environment. In traditional small scale computer systems, more than one task shares a processor in a time-multiplexed way: when a task waits for an event which may occur in the far future, it renders the processor to others so that the shared processing facility could be better utilized to increase the overall system performance. To increase the processing capability, more processors are used in the time-multiplexed computing environment along with similar increases in other resources such as main memory, enabling concurrent processing of tasks from the same problem. Techniques for handling routing deadlock are not sufficient for dealing with message-dependent deadlocks because they assume that messages in the network always sink upon arrival at their destinations. That is, it is assumed that the delivery of messages is not coupled in any way to the injection (generation) or reception (consumption) of any other message in the network or at network endpoints.

1.1 Interconnection network

Interconnection networks are built with a set of communication links, routers (or switches), and network interfaces attached to end nodes. Communication links distribute packets among routers and network interfaces. End nodes use a network interface through which they gain access to one or more adjacent routers or other end nodes. Network interfaces usually provide buffer resources in order to temporarily hold input and output packets in the event of network congestion or delays in processing. The links connecting a network interface and a router are called injection channels and delivery channels, depending upon the direction of packet flow.

1.2 Efficient use of interconnection network

The main objective of deadlock handling techniques is to prevent network resources from being blocked forever without making progress in packet delivery. However, such techniques heretofore provide no way of accelerating the movement of packets already involved in congestion or deadlock. These approaches aid in achieving high network performance, but may not be enough. What is needed is the development of mechanisms that address the problem at its source: namely, the blocked packets—whether they are within the network or at network endpoints. Under certain circumstances of network loading, traffic bursts and pathological communication behavior, blockage of a single resource can easily grow into paths of blockage dependencies, cyclic blocking, and even knotted cyclic blocking across many network resources by packets.

2. Methodology

The technique proposed for handling message-dependent deadlock is derived from a progressive deadlock recovery technique proposed previously for handling routing-dependent deadlocks, called Disha Sequential [22]. In Disha Sequential (or simply Disha), each router has a centralized flit-sized deadlock buffer (DB) used to progressively route potentially deadlocked messages once detected. Only one message is allowed to use the set of deadlock buffers at any given time, implemented by a token passing and capture mechanism. A circulating token visits all nodes, is captured by a node containing a potentially deadlocked message, and is released once the message reaches its destination. With this mechanism, the routing function defined on the set of deadlock buffers provides a connected and deadlock-free recovery path to/from any two network endpoints. The Disha Sequential scheme is applicable to all networks and is proven to safely recover from all potential routing-dependent deadlocks. The proposed technique, extends the notion of Disha recovery paths existing only between network endpoints to one that includes network endpoints as well. Hence, the circulating token must also visit all network interfaces attached to each router node, and a deadlock buffer—referred to as a deadlock message buffer or DMB—must also be provided in each network interface. The size of the DMB is determined by the minimum unit of information on which end-to-end error detection/protection (i.e., ECC, checksum) is performed. Typically, this is at the packet level, requiring these deadlock buffers to be at least packet-sized. This is also the minimum size of the network interface input and output message queues; however, larger input/output queues are typically used to increase performance.

3. Results

3.1 Figures

The diagram 1 shows that each token receiver returns the token to the associated token sender. If the memory controller of a token receiver generates multiple subordinate messages for the message type being processed (i.e., $m_i < m_{j1}, m_{j2} ...m_{jn}$ as would be the case for a multicast/broadcast transaction such as invalidate), the node repeatedly uses the token to deliver each subordinate message m_{jm} $(1 \le m \le n)$ to its destination in turn before returning the token to that node's sender. All potentially deadlocked messages that undergo this progressive recovery procedure are said to be rescued. If the token returns to the node which initiated the recovery process and the node has no more messages to deliver with the token, the deadlock recovery process ends. At this point, the token is released for re-circulation.

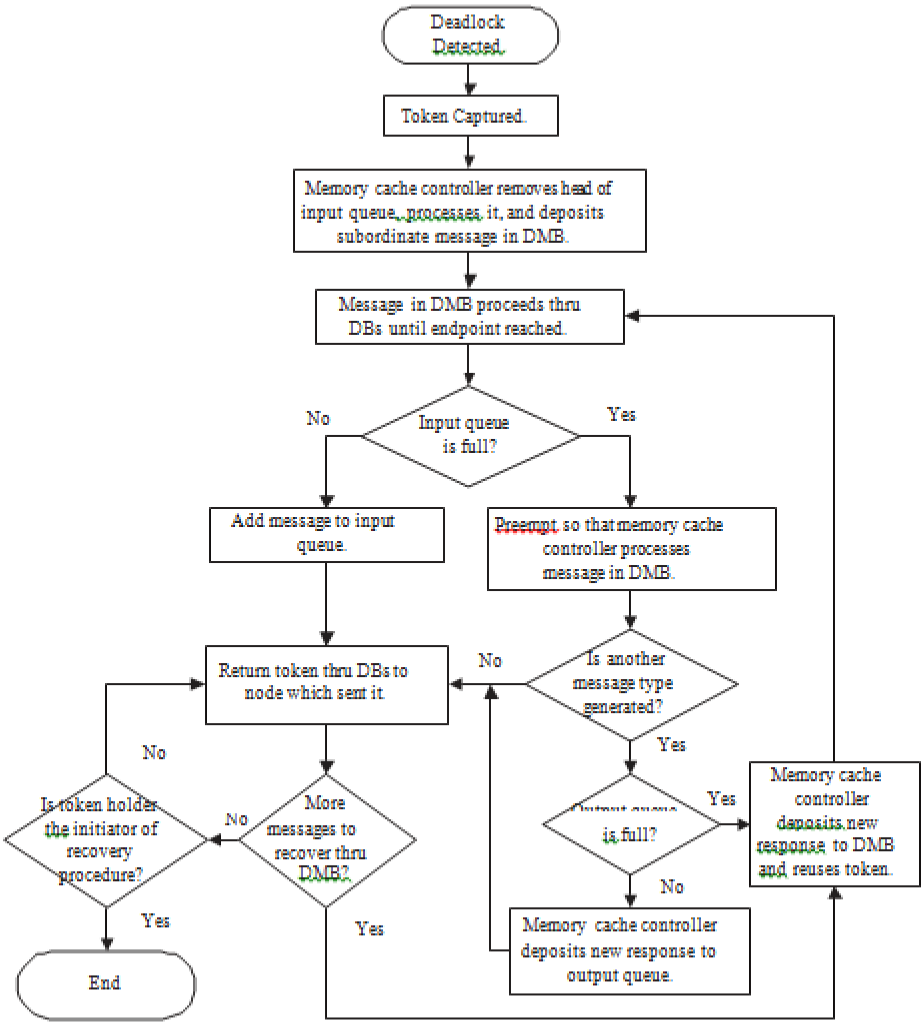

Figure 1: Deadlock Recovery

3.2 Result

The implementation of JDK1.5 is done on the UDT protocol with security to remove the deadlock in the network connected in Ring Topology

4. Discussion

Objective of this paper is to study various transport protocols and finding of one protocol which can transport data at high speeds like 2 tetra bytes. TCP variants and new protocols such as XCP suffer from deployment difficulties, application level solutions emerge as a favorite timely solution. A common approach is to use parallel TCP, such as PSockets and Grid FTP. Using multiple TCP flows may utilize the network more efficiently, but this is not guaranteed. Performance of parallel TCP relies on many factors from end hosts to networks. For example, the number of parallel flows and the buffer sizes of each flow have significant impact on the performance. The optimal values vary on specific networks and end hosts and are hard to tune. In addition, parallel TCP inherits the RTT fairness problem of TCP.

5. Conclusion

In parallel computer systems, the efficiency of interconnection networks has a significant impact on the performance of the overall system. At the same time, the emergence of network-based applications emphasizes the need for high performance networks. One of the common approaches for improving network performance is to handle deadlocks by recovering from them whenever detected. A recovery approach allows channel resources to be exploited in a less restrictive way and, thus, requires less channel resources for achieving the same performance as a deadlock avoidance approach. This research presents new and superior mechanisms for handling network congestion and deadlock as a way to improve network utilization and efficiency.

References

[1] Andrew A.,1992:' Planar-Adaptive Routing: Low-Cost Adaptive Networks for Multiprocessors', *IEEE Computer Soci- ety*,**1**,268-277.
[2] Baydal E, Lo`pez P.,Duato J.,2000:' A Simple and Efficient Mechanism to Prevent Saturation in Wormhole Networks.', *14th International Conference on Parallel and Distributed Processing Symposium*, 617-622.
[3] Chien A. A.,1998:' A cost and speed model for k-ary n-cube wormhole routers', *IEEE Transactions on Parallel and Distributed Systems*,**2**,150–162,
[4] Caminero B., Carrion C., Quiles F.J., Duato J., and Yalamanchili S.,2002:'Investigating Switch Scheduling Algorithms to Support QoS in the Multimedia Router.', *Workshop on Communication Architecture for Clusters (CD- ROM).*
[5] Censier L. M., Feautrier P.,1978:'A New Solution to Coherence Problems in Multicache Systems.', *IEEE Transactions on Computers*, C-27:1112–1118.
[6] Clark D. D., Pogran K. T., Reed D. P., 1978:'An Introduction to Local Area Networks.',*IEEE*, 66(11),1497–1517.
[7] Dally W., Seitz C.,1987:,'Deadlock-free Message Routing in Multiprocessor Interconnection Networks.',*IEEE*, 5,547–553.
[8] Duato J.,1993:,'A New Theory of Deadlock-free Adaptive Routing in Wormhole Net- works.',*IEEE Transactions on Parallel and Distributed Systems*, 4,1320–1331.
[9] Eijk97 Van P.,1997:,'Gigabit Ethernet: Technology, Systems, and Network Applications.', *Electronic Design*, **7**,:64–73.
[10] Fischer W., Wallmeier E., Worster T., Davis S.P., Hayter A, 1994:'Data Com- munications using ATM: Architectures, Protocols, and Resource Management.', *IEEE Communications Magazine*, 24–33.