A Necessary and Sufficient Condition for Deadlock-Free Message Routing in Communication Networks

Elavarasi, G. Raja
Department of Computer Science Engineering, Dhanalakshmi Srinivasan Engineering College, Perambalur, Tamil Nadu, India

Article Info
Article history:
Received 20 March 2015
Received in revised form
10 April 2015
Accepted 20 May 2015
Available online 15 June 2015

Keywords
Deadlocks,
Routing Algorithm,
Communication Networks,
Wormhole Switching

Abstract

Deadlocks are an important issue in the design and analysis of communication networks. Wormhole switching is a popular switching technique in direct networks. It refers to a simple flow control system in computer network that is primarily based on fixed links. It also reduces the latency and storage requirements on each node. Deadlock analysis of routing function is a manual and complex task. In the absence of contention, latencies are proportional to the sum of the packet length and the distances to travel. We propose an algorithm to analyze the deadlock in communication networks. The deadlock-free routing algorithm is the first to automatically check a necessary and sufficient condition for deadlock-free routing. Our algorithm performs Effective analysis in this network.

1. Introduction

A computer network or data network is a telecommunications network that allows computers to exchange data. In computer networks, networked computing devices pass data to each other along data connections. Data is transferred in the form of packets. The connections (network links) between nodes are established using either cable media or wireless media. The best-known computer network is an internet. Network computer devices that originate, route and terminate the data are called network nodes. Two such devices are said to be networked together when one device is able to exchange information with the other device, whether or not they have a direct connection to each other. In most cases, communications protocols are layered on (i.e. work using) other more specific or more general communications protocols, except for the physical layer that directly deals with the physical media.

A computer network consists of a collection of computers, printers and other equipment that is connected together so that they can communicate with each other. There are two types of network configuration, peer-to-peer networks and client/server networks. Peer-to-peer networks are more commonly implemented where less than ten computers are involved and where strict security is not necessary. All computers have the same status, hence the term 'peer', and they communicate with each other on an equal footing. Files, such as word processing or spreadsheet documents, can be shared across the network and all the computers on the network can share devices, such as printers or scanners, which are connected to any one computer.

Client/server networks are more suitable for larger networks. A central computer, or 'server', acts as the storage location for files and applications shared on the network. Usually the server is a higher than average performance computer. The server also controls the network access of the other computers which are referred to as the 'client' computers.

Corresponding Author,
E-mail address: elavarasi.visu@gmail.com

A communications protocol is a set of rules for exchanging information over network links. In a protocol stack (also see the OSI model), each protocol leverages the services of the protocol below it. An important example of a protocol stack is HTTP running over TCP over IP over IEEE 802.11. (TCP and IP are members of the Internet Protocol Suite. IEEE 802.11 is a member of the Ethernet protocol suite.)

Whilst the use of protocol layering is today ubiquitous across the field of computer networking, it has been historically criticized by many researchers for two principal reasons. Firstly, abstracting the protocol stack in this way may cause a higher layer to duplicate functionality of a lower layer, a prime example being error recovery on both a per-link basis and an end-to-end basis. Secondly, it is common that a protocol implementation at one layer may require data, state or addressing information that is only present at another layer, thus defeating the point of separating the layers in the first place. For example, TCP uses the ECN field in the IPv4 header as an indication of congestion; IP is a network layer protocol whereas TCP is a transport layer protocol.

Communication protocols have various characteristics. They may be connection-oriented or connectionless, they may use circuit mode or packet switching, and they may use hierarchical addressing or flat addressing. Ethernet is a family of protocols used in LANs, described by a set of standards together called IEEE 802 published by the Institute of Electrical and Electronics Engineers. It has a flat addressing scheme. It operates mostly at levels 1 and 2 of the OSI model. For home users today, the most well-known member of this protocol family is IEEE 802.11, otherwise known as Wireless LAN (WLAN or WiFi). The complete IEEE 802 protocol suite provides a diverse set of networking capabilities.

The Internet Protocol Suite, also called TCP/IP, is the foundation of all modern networking. It offers connection-less as well as connection-oriented services over an inherently unreliable network traversed by data-gram transmission at the Internet protocol (IP) level. At its core, the protocol suite defines the addressing, identification, and routing specifications for Internet Protocol Version 4 (IPv4) and for IPv6, the next generation of the protocol with a much enlarged addressing capability.

2. Related Work

In concurrent programming, a deadlock is a situation in which two or more competing actions are each waiting for other to finish, and thus neither ever does. Mutual exclusion, Hold and Wait, No Pre-emption, Circular wait are the characteristics of the Deadlock. Mutual exclusion means one or more than one resource must be held by a process in a non-sharable (Exclusive) mode. If a process holds a resource while waiting for another resource is called Hold and Wait. No Pre-emption means there is only voluntary release of a resource - nobody else can make a process give up a resource. If Process A waits for Process B waits for Process C ....waits for Process A is called Circular Wait. Ensure deadlock never occurs using either prevent any one of the 4 conditions from happening. Avoidance means Allow all deadlock conditions, but calculate cycles about to happen and stop dangerous operations. Allow deadlock to happen. This requires using both detection and recovery. Detection means know a deadlock has occurred. Recovery means regain the resources.

If we have prior knowledge of how resources will be requested, it's possible to determine if we are entering an unsafe state. Possible states are deadlock, unsafe state and safe state. Deadlock means no forward progress can be made. If a state that may allow deadlock is called unsafe state. Safe state means state is safe if a sequence of processes exist such that there are enough resources for the first to finish, and as each finishes and releases its resources there are enough for the next to finish.

Many recent experimental and commercial parallel computers use direct networks for low latency, high bandwidth inter processor communication. A typical direct network is the k-ary n-cube network, which has an n-dimensional grid structure with k nodes (processors) in each dimension such that every node is connected to two other nodes in each dimension by direct communication links. The performance of a multicomputer network depends on the switching technique and the routing algorithm used. Possible switching techniques are the virtual cut-through, store-and-forward, and wormhole. The wormhole (WH) switching technique has been widely used in the recent multi computers. In the WH technique, a packet is divided into a sequence of fixed-size units of data, called flits. If a communication channel transmits the first flit of a message, it must transmit all the remaining flits of the same message before transmitting flits of another message. The main advantages of wormhole switching are low memory requirements in routers and pipelined data movement in the absence of contention. The main disadvantage of wormhole switching is channel congestion, since a blocked message does not relinquish the communication channels it has already acquired. The virtual cut-through; VCT, and store-and-forward, SAF, switching techniques require more storage in nodes but have less channel contention. Some of the most important issues in the design of a routing algorithm are high throughput, low-latency message delivery, and avoidance of deadlocks, live locks, and starvation. In this study we consider only minimal routing algorithms as per which a message always moves closer to its destination with each hop taken. Live locks can be avoided with minimal routing, and starvation can be avoided by allocating resources such as communication channels and buffers in FIFO order. Ensuring deadlock freedom depends on the design of the routing algorithm.

2.1. System Design

Existing System Using the header flit that contains the routing information. The remaining flits consist of the payload and a tail flit indicating the end of the packet. Only the header flit is used for routing. The data and tail flits follow the header flit in a pipelined fashion. Wormhole switching provides low message latency and requires small buffer capacity in the channels of the network as buffers need not store whole packets. This switching technique is also prone to deadlock. The flits of a single packet often hold many resources simultaneously, blocking many other messages. Deadlocks are a key issue in the design of wormhole networks. Duato’s methodology has been very popular and supported the design of very efficient routing protocols. But, it is a manual process. It is error-prone and not scalable. If a large packet is being sent, the destination of the packet is not ready to receive it. Something has gone wrong on the network, (e.g) A link failure, so that a packet cannot move forward across the failed link.

A network and a routing function that are already known to be deadlock-free. New channels and routing capabilities can be added to this network as long as once a message arrives in an original channel, it cannot be routed towards new channels. Duato’s condition holds for any network obtained in this way. The set of networks that can be proven deadlock-free using Duato’s theorem is much larger than the set of networks that can be obtained using his design methodology. However, using Duato’s theorem to show absence of deadlock of a network where his design methodology does not apply can require a highly complicated manual proof.

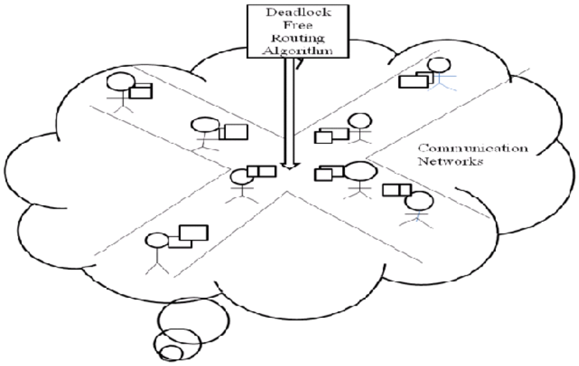

Fig: 1. Architecture Diagram

We proposed the first necessary and sufficient algorithm for deadlock verification in wormhole networks. Our algorithm only requires as input a description of the routing function. The only assumption on the routing logic is that it is memory less, i.e., it decides the routes based on the current position of the message and the destination. Our decision procedure decides whether this routing function satisfies Verbeek and Schmaltz’ necessary and sufficient condition. It returns, whenever the network is not deadlock-free, an exact description of the deadlock. This entails which messages participate, which channels they occupy, and for which destination they are headed. To ensure that this feedback is readable, we return the smallest deadlocks that can occur. Due to various optimizations, our approach scales to sufficiently large networks.

2.2. Waypoint Connectivity Construction

Waypoint connectivity construction is a network topology in which each node relays data for the network. All nodes cooperate in the distribution of data in the network. When using a routing technique, the message is propagated along a path, by hopping from node to node until the destination is reached. To ensure all its paths' availability, a routing network must allow for continuous connections and reconfiguration around broken or blocked paths, using self-healing algorithms.

2.3. Checking Sufficient Condition

Checking sufficient condition is the first necessary and sufficient algorithm for deadlock verification in wormhole networks. Our algorithm only requires as input a description of the routing function. The only assumption on the routing logic is that it is memory less, i.e., it decides the routes based on the current position of the message and the destination. Our decision procedure decides whether this routing function satisfies necessary and sufficient condition.

2.4. Detect and Prevent Deadlocks

Deadlock detection is the process of actually determining that a deadlock exists and identifying the processes and resources involved in the deadlock. The basic idea is to check allocation against resource availability for all possible allocation sequences to determine if the system is in deadlocked state a. Of course, the deadlock detection algorithm is only fully of this strategy. Once a deadlock is detected, there needs to be a way to recover several alternatives exists: Temporarily prevent resources from deadlocked processes. Back off a process to some check point allowing pre-emption of a needed resource and restarting the process at the checkpoint later; successively kill processes until the system is deadlock free.

2.5. Checking Complete Status

Checking complete status is the final phase. Here checking and analyzing the whole network and then each and every state to check the user status.

3. Conclusion

Wormhole switching is a complex switching technique. Designing adaptive routing functions for wormhole networks is a difficult task. The routing logic satisfies a complex condition for deadlock-free routing. The first algorithm which automatically decides absence of deadlocks in wormhole networks. It provides readable feedback in case the network is not deadlock-free. The algorithm is sound and complete.

References

[1] W. J. Dally, C. Seitz, Deadlock-Free Message Routing in Multiprocessor Interconnection Networks, IEEE Trans. Comput., 36(5), 1987, 547-553.

[2] J. Duato, A Necessary and Sufficient Condition for Deadlock-Free Adaptive Routing in Wormhole Networks, IEEE Trans. Parallel Distrib. Syst., 6(10), 1995, 1055-1067

[3] J. Duato, A New Theory of Deadlock-Free Adaptive Routing in Wormhole Networks, IEEE Trans. Parallel Distrib. Syst., 4(12), 1993, 1320-1331.

[4] C. Grecu, Rusu, A Flexible Network-on-Chip Simulator for Early Design Space Exploration, IEEE transactions on Microsystems and Nano Electronics Research Conference, 7(2), 2008

[5] C. J. Glass, L. M. Ni, The Turn Model for Adaptive Routing, J. ACM, 41(5), 1994, 874-902

[6] M. Kaufmann, P. Manolios, J. S. Moore, ACL2 Computer-Aided Reasoning: An Approach. Dordrecht, The Netherlands: Kluwer, 2011

[7] W. Luo, D. Xiang, An Efficient Adaptive Deadlock-FreeRouting Algorithm for Torus Networks, IEEE Trans. Parallel Distrib. Syst., 23(5), 2012, 800-808

[8] Sheng Ma, Natalie Enright Jerger, Whole Packet Forwarding: Efficient Design of Fully Adaptive Routing Algorithms for Networks-on-Chip, IEEE International Symposium on High Performance Computer Architecture, 2012

[9] F. Verbeek, J. Schmaltz, A Comment on ‘A Necessary and Sufficient Condition for Deadlock-Free Adaptive Routing in Wormhole Networks, IEEE Trans. Parallel Distrib. Syst., 22(10), 2011, 1775-1776

[10] F. Verbeek, J. Schmaltz, On Necessary and Sufficient Conditions for Deadlock-Free Routing in Wormhole Networks, IEEE Trans. Parallel Distrib. Syst., 22(12), 2011, 2022-2032

[11] F. Verbeek, J. Schmaltz, Automatic Verification for Deadlock in Networks-on-Chips with Adaptive Routing and Wormhole Switching, IEEE NOCS, 2011, 25-32

[12] F. Verbeek, J. Schmaltz, Formal Verification of a Deadlock Detection Algorithm, 10th Int. Workshop ACL2 Theorem Prover Appl., 2011, 103-112

[13] Zaheer Ahmed, A Fault Adaptive Routing, J. ACM, 41(5), 2009, 874-902.

[14] Zhonghai Lu, Layered switching for networks on chip, IEEE Trans. Parallel Distrib. Syst., 22(10), 2011, 1775-1776