제 18 회 한국정보처리학회 추계학술발표대회 논문집 제 9 권 제 2 호 (2002.11)

채널 기반 인피니밴드의 서브넷 관리를 위한 시뮬레이션

김영환*, 윤희용*, 박창원**,이형수**, 고재진**, 박상현**
*성균관 대학교 정보통신 공학부
**전자부품연구원 정보시스템 연구 센터
{yhkim93, youn}@ece.skku.ac.kr, {parkcw,hslee,jaejini,shpark}@keti.re.kr

Simulation of Subnet Management for InfiniBand

Young Hwan Kim*, Hee Yong Youn*, Chang Won Park**, Hyoung Su Lee**, Jae Jin Go**,
Sang Hyun Park**
*School of Information and Communication Engineering
SungKyunKwan University
**IT System Research Center, Korea Electronics Technology Institute

Abstract

InfiniBand is a switched-fabric architecture for next generation I/O systems and data centers. The InfiniBand Architecture (IBA) promises to replace bus-based architectures, such as PCI, with a switched-based fabric whose benefits include higher performance, higher RAS (reliability, availability, scalability), and the ability to create modular networks of servers and shared I/O devices. The switched-fabric InfiniBand consists of InfiniBand subnets with channel adapters, switches, and routers. In order to fully grasp the operational characteristics of InfiniBand architecture (IBA) and use them in ongoing design specification, simulation of subnet management of IBA is inevitable. In this paper, thus, we implement an IBA simulator and test some practical sample networks using it. The simulator shows the flow of operation by which the correctness and effectiveness of the system can be verified.

Key words: InfiniBand, network management, simulation, subnet, switch fabric.

1. Introduction

Almost all computing equipments are linked to some networks due to fast development of network technology and efficient information sharing. Internet or LAN, WAN etc. is no longer a selective item and connecting high-capacity high-speed storage to the network is a recent trend. The extent of such connection is being magnified rapidly by home networking that connects home PCs, electronic appliances, and storages to efficiently share and manage data. Therefore, the necessity of efficient examination and management of the state of the whole network and equipments is rising. It is especially important for channel-based InfiniBand used between high-capacity storage and server I/O Architecture, ranging from a small server with one processor and a few I/O devices to a massively parallel supercomputer with hundreds of processors and thousands of I/O devices.

Now, there are several architectures developed for managing the nodes in a network, and they are referred as network management system (NMS). Generally, a network management system contains two primary elements: a *manager* and *agents*. The Manager is the console through which the network administrator performs network management functions. The Agents are the entities that interface to the actual devices being managed. Switches, hubs, routers, or network servers including scaleable storages are examples of managed devices containing some objects managed. The objects might be hardware, configuration parameters, performance statistics, and so on, that directly relate to the current operation of the device in question. They are arranged in what is known as a virtual information database called a management information base (MIB). SNMP allows the managers and agents to communicate for the purpose of accessing these objects. In InfiniBand fabric, the method used for managing the nodes also is similar to that with SNMP.

InfiniBand has been recently proposed as a standard for communication between processing nodes and I/O devices as well for inter-processor communication. To manage an InfiniBand fabric, two entities are defined - Subnet Manager (SM) and Subnet Management Agent (SMA). Only one subnet manager is needed per subnet and it can reside in any node including switches and routers. Subnet manager uses a special class of Management Datagram (MAD) called a Subnet Management Packet (SMP) which is directed to a special queue pair (QP0).

In order to fully grasp the operational characteristics of InfiniBand architecture (IBA) and use them in ongoing design specification, simulation of IBA is inevitable. In this paper, thus, we implement an IBA simulator and test some practical sample network using it. The simulator shows the flow of operation by which the correctness and effectiveness

of the system can be verified.

The rest of the paper is organized as follows: the next section gives InfiniBand Architecture overview. In Section 3 we present a simulation model and the result of subnet management in InfiniBand. Finally, in Section 4 we conclude the paper with a summary.

2. Infiniband

2.1 Overview

The InfiniBand Architecture (IBA) specification describes a Storage Area Network (SAN) connecting multiple independent processor platforms (i.e., host processor node), I/O platforms, and I/O devices. The architecture is independent of the host operating system and processor platform. IBA is designed around a switch-based interconnect technology with high-speed point-to-point links. An IBA network is divided into subnets interconnected by routers, where each subnet consists of one or more switches, processing nodes, and I/O devices. Routing in IBA subnet is distributed, based on forwarding tables stored in each switch. IBA supports any topology defined by the user, including irregular ones in order to provide flexibility and incremental expansion capability.

IBA links are full-duplex point-to-point communication channels. The signaling rate on the links is 2.5 GHz. Physical links may be used in parallel to achieve greater bandwidth. Currently, IBA defines three link bit rates. The lowest one is 2.5Gbps and is referred to as 1x. Other link rates are 10Gbps (referred to as 4x) and 30Gbps (referred to as 12x) that correspond to 4-bit wide and 12-bit wide links, respectively. The width or widths that will be supported by a link is vendor-specific.

IBA switches route messages from their source to destination based on forwarding tables programmed with forwarding information during initialization and network modification. Messages are segmented into packets for transmission on links and through switches. The packet size is such that after headers are considered, the Maximum transfer Unit (MTU) of data may be 256 bytes, 1KB, 2KB or 4KB. The interested reader is referred to the InfiniBand specification for more details [1-4].

2.2 Subnet Management

Each subnet has at least one subnet manager (SM). SM resides on a port of a channel adaptor, router, or switch, and can be implemented either in hardware or software. When there are multiple SMs on a subnet, one SM will be the master SM amd the remaining SMs become standby SMs.

The master SM is a key element in initializing and configuring an IB subnet. The master SM is elected as part of the initialization process for the subnet and is responsible for:

- Discovering the physical topology of the subnet
- Assigning Local Identifiers (LIDs) to the endnodes, switches, and routers
- Establishing possible paths among the endnodes
- Sweeping the subnet to discover and manage topology changes as nodes are added and deleted.

Figure 1 shows a subnet in channel-based InfiniBand, which consists of host with processor, I/O device (i.e. storage) and switch.

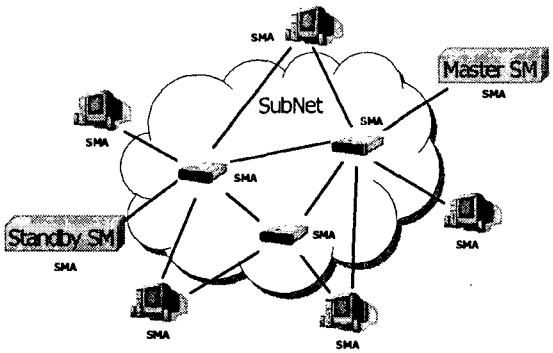

Figure 1. Subnet architecture.

The communication between the master SM and SMAs, and among the SMs is performed with subnet management packets (SMPs).

2.2.1 Subnet Manager (SM)

There may be one or more SMs operating on a subnet. Each SM indicates its presence on the subnet by setting the *IsSM* bit of *PortInfo:CapabilityMask* on the port where it resides.

Each SM is always in a particular state - Master, Standby, Discovering, or Not-active. An SM shall comply with the state machine shown in Figure 2 during its startup and become either a master or standby on the subnet. Furthermore, the state machine specifies how a single Master SM is maintained during subnet topology changes, packet loss, addition/removal of SMs, and subnet merges.

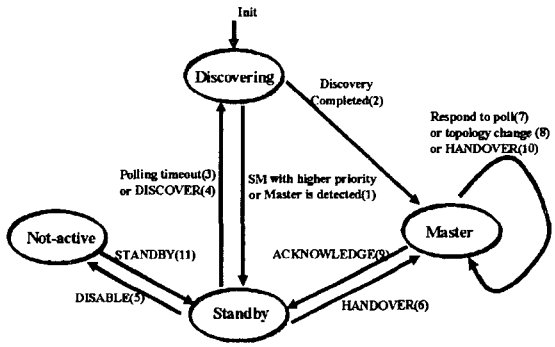

Figure 2. The state machine for an SM.

- Discovery State: the initial state. At startup, an SM enters the DISCOVERING state. In the DISCOVERING state, the SM performs repetitive *SubnGET(*)* to find all the nodes and SMs on the subnet. Also, collect many other attributes during discovery.
- Standby State: standby SMs shall not configure the subnet. Each Standby SM polls the Master SM with *SubnGet(SMInfo)*SMPs. As long as a Standby SM determines that the Master SM is alive, it stays in *SMInfor:SMState*=STANBY.
- NotActive State: if an SM is in the NOT-ACTIVE state, it shall not send SubnSet() or SubnGet() SMPs but reply to SubnSet(SMInfo) and SubGet(SMInfo)

SMPs.

- Master State: the Master SM starts its operation by topology discovery, LID verification and assignment, path verification and calculation, etc.

2.2.2 Subnet Management Agent (SMA)

Each channel adaptor (CA) and router, and switch will have a Subnet Management Agent (SMA) that communicates with the Subnet Management Interface (SMI) which abstractly represents a target to which messages may be sent and through which messages will be processed or will be dispatched to an appropriate processing entity. For management interfaces, the associated processing entity is an agent or, in some cases, a manager. As such, an interface is a means to gain access to the functionality of agents and/or managers and SM. The SMA will respond and generate SMPs. The requirements of SMA behavior are following:

- SubnGet: An SMA may receive a SMP from the subnet containing a SubnGet at any time. An SMP containing a SubnGetResp is returned according to the rules.
- SubnSet: The SMA updates the registers appropriate with the contents of the attribute contained in the SMP. Then an SMP containing a SubnGetResp is returned.
- SubnGetResp: the SMA generates SubnGetResp, then it fills the attribute identified in the request with the appropriate contents of register information. After transmission of the response, the SMA discards any residual state associated with that SMP.
- SubnTrap: Traps may be issued to report an event in any port on the subnet.

2.2.3 Subnet Management Packet (SMP)

Management Datagrams (MADs) are the basic elements of the massaging scheme defined for communication management. These MADs are also referred to as Subnet Management Packets (SMPs). Fgure 3 shows the base format of SMP. The Method is used to perform based on the management class (i.e. SubnGet(), SubnSet(), SubnGetResp(), SubnTrap()).

bytes					
0	BaseVersion	MgmtClass	ClassVersion	R	Method
4	Status		ClassSpecific		
8	TransactionID				
12					
16	AttributeID		Reserved		
20	AttributeModifier				
24	Data				
...					
252					

Figure 3. MAD format.

There are two types of SMPs: LID routed and directed route.

- LID routed SMP: LID routed SMPs are forwarded through the subnet based on the LID of the destination and using the normal switch forwarding tables set up during subnet initialization [8].
- Directed route SMP: Directed route SMPs are forwarded to the subnet based on a vector of port numbers that define a path through the subnet. Directed route SMPs are used to implement several management functions, in particular, before the LIDs are assigned to the nodes. Also, Directed route SMPs are used to route between SMAs using a store-and-forward technique between neighboring nodes. They are therefore not dependent on routing table entries. It is primarily used for discovering the physical connectivity of a subnet before it has been initialized [8].

2.2.4 SMP Interface

Two required interfaces to management entities are specified based upon two well known queue pairs (QP0). These are known as the Subnet Management Interface (SMI) and General Service Interface (GSI). The SMI is associated with QP0, which is used exclusively for sending and receiving SMPs. Communications with the SMA is always through the SMI. If a channel adapter, switch, or router hosts an SM, then communications between that SM and the SMAs in the subnet is also through the SMI. Figure 4 shows the position of SMI.

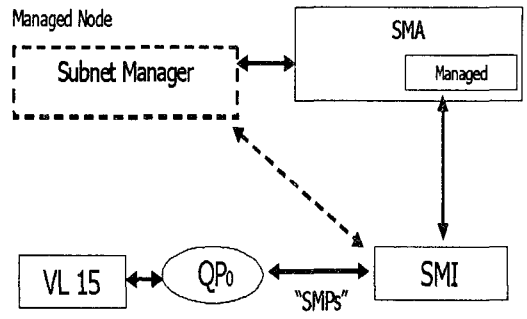

Figure 4. SMP Interface.

Messages arriving at QP0 are processed. QP0 has unique semantics with respect to message processing while specifying one of them as the destination queue pair. Implementation of QP0 is not required to follow the semantics associated with other queue pairs with respect to requirements such as posting and consumption of WEQs, manipulation of an associated completion queue, and so on.

3. Simulation

The simulator is developed based on IBTA's Infiniband Specification release 1.0a [1]. The purpose of the simulator is to simulate the establishment of a subnet with an InfiniBand fabric concentrating on the subnet manager activities, which are subnet discovery, master SM determination and configuration. Management models defined in the simulator are QP, SMI, SMA, and SM. The features of the simulator are as follows:

- Defines a subnet topology consisting of switches and end-nodes interconnected by links. Each new com-

ponent is configured with its relevant properties. The topology can be changed by addition or removal of a physical component.

- Runs a simulation on a defined topology. Since the emphasis is on subnet management, communication between CAs (that actually represents the achievement of the subnet) is not supported.
- Enables the user to change topology, especially adding and removing links.

Figure 5 shows a sample subnet topology simulated by the simulator developed. It consists of seven hosts - a processor, four I/O devices (i.e. storage) and two switches. Observe that there are one master SM and two standby SMs.

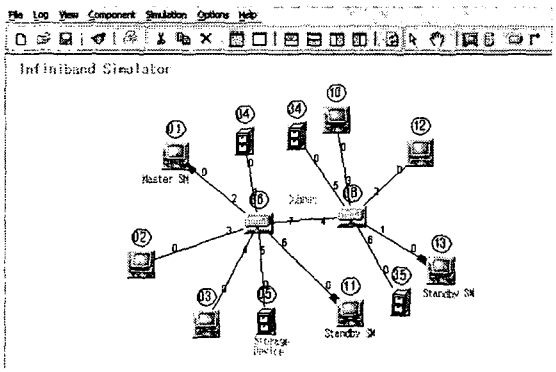

Figure 5. A sample topology simulated.

For simulating the sample topology above, six classes are required: QP, SMI, SMA, SM, Switch, and Link. Their operations are as follows.

- QP Class: General implementation of this class is used to define SMI and GSI.
- SMI Class: The SMI class is known as QP0 (Management QP). Within the implementation the SMI and QP0 are two entities as identified in IBTA's Infiniband Specification release 1.0a. The SMI implementation supports both subnet establishment and regular (post-establishment) operation, handling directed-routed and LID-routed management and data packets.
- SMA Class: SMA class is made to provide the most general implementation. Like the fabric, the SMA is designed to work in passive mode during and after subnet establishment and can be enhanced easily.
- SM Class: Due to complexity, we initially implement only the subnet establishment. Also, The SM state machine is implemented as required by the client.
- Switch Class: The switch class has a management port, port number 0, that holds management entities (SMA, SMI,..) that all the other ports attached to it use.
- Link Class: Link class implements an IB link that connects two ports.

Figure 6 shows the part of simulation result. We can see that SM1, the master SM, sends and receives SMPs to every other node for managing the subnet.

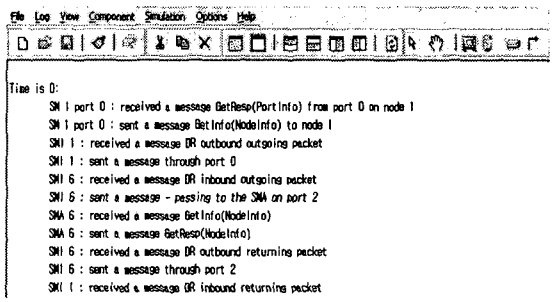

Figure 6. Simulation result.

In order to obtain sensible results, the simulator can also be implemented using multi-threading, enabling each component on the subnet to run concurrently.

4. Conclusion

In this paper we have presented the concept of subnet management in channel-based InfiniBand, which is next generation I/O system. Recently, InfiniBand Specification final release 1.0a was released. Since the current specification is not optimal yet, more research is needed. A comprehensive Infiniband simulator implementation may require considerable development and design effort. Moreover, certain issues need to be solved in order to prevent undefined or irrational behavior of the simulation.

Reference

[1] InfiniBand Architecture release 1.0a Volume1 –General Specification.
http://www.infinbadta.org
[2] Lane15 Channel Interface Specification January 2002
http://www.lane15.com
[3] Gregory F.PFISTER, "An Introduction to the Infiniband Architecture"
[4] Ramon D. Acosta "Introduction to the InfiniBand Interoperability Challenges"
http://www.lane15.com
[5] InfiniBand Simulator
http://203.252.46.130/infiniband.htm
[6] InfiniBand™ Trade Association.
[7] G. Pfister. High Performance Mass Storage and Parallel I/O, chapter 42: "An Introduction to the InfiniBand Architecture", pages 617-632. IEEE Press and wiley Press, 2001.
[8] Sancho, J.C.; Robles, A.; Duato, J "Effective strategy to compute forwarding tables for infiniband networks" Parallel Processing, International Conference on, 2001. , 2001 Page(s): 48 -57