Chapter 1
Basic Terms and Concepts
第 1 章
基本术语和概念

This Chapter | 这一章

This chapter provides 本章提供

- Advantages of the InfiniBand network architecture. IB 网络架构(体系结构)的优势所在。
- An introduction to basic terminology. 基本术语介绍。
- Packet addressing basics. 分组(包)寻址基础。
- The basic roles of channel adapters, routers, switches, and repeaters. 通道适配器(CA)， 路由器，交换机和中继器(转发器)的基本作用(角色)。
- An introduction to message passing. 报文(消息)传递介绍。

The Next Chapter | 下一章

The next chapter introduces the concept of device attributes, managers, management agents (MAs), and management datagrams (MADs). 下一章将介绍一些基本概念，包括设备属性，管理器，管理代理和管理数据报。

1.1 Definition of the Acronym "IBA" | 缩写词"IBA"的定义

In the InfiniBand specification, the acronym IBA stands for InfiniBand Architecture. 在 IB 的规范中，缩写词 IBA 代表 IB 架构(体系结构)。

1.2 Packet Field Documentation Convention | 包字段文档约定

Many of the fields contained in a packet are comprised of subfields. When the specification refers to a subfield, it is frequently represented as:

FieldName:SubFieldName.

As an example, LRH:SL is referring to the Service Level subfield within the packet's Local Route Header field. 在一个数据包(分组)中，很多字段都是有多个子字段组成。当 IB 的规范中提及一个子字段的时候，子字段经常地被表示为:

字段名称：子字段名称

例如，LRH:SL 指的是在一个数据包(分组)中的本地路由报头(LRH)字段里的服务级别(SL)子字段。

1.3 InfiniBand Advantages | IB 的优势所在

Some of the major advantages offered by InfiniBand network architect-ure include the following: IB 网络架构提供的主要优势体现在如下几个方面：

- Supports multiple protocols, including tunneling packets associated with virtually any non-InfiniBand protocol through an InfiniBand fabric. 支持多种协议，几乎囊括所有能够穿透 IB fabric 的，与任何非 IB 协议相关联的隧道数据包。
- Throughput of 2.5Gb/s, 10Gb/s, or 30Gb/s is achievable (depending on the link width implemention). 在吞吐量方面可以达到 2.5Gb/s, 10Gb/s, 或 30Gb/s。(当然，具体吞吐量取决于链路的宽度实现。)
- Non-privileged applications can send and receive messages without causing a kernel privilege mode switch. 对那些不享有特权的应用程序而言，无需卷入内核特权模式切换，就能够发送和接收消息(报文)。
- The processor is not involved in message passing. Rather, each network message transfer is handled by hardware DMA transfer engines within the channel adapter. 处理器并不参与任何消息传递。相反，位于通道适配器 (CA) 内部的 DMA 传输引擎全权负责处理任何一个网络消息的传输。
- The InfiniBand protocol includes message transfer commands that permit direct memory-to-memory message transfers between the local memories of two channel adapters. IB 协议中包括一组消息传输命令，这组命令允许执行"内存到内存"的消息传输，这里的"内存"特指通道适配器的局部内存(后面一律翻译为**"CA** 卡内存**"**)。
- The majority of the protocol layers can be implemented in silicon, thereby minimizing the burden placed on software and the processor. 大多数协议层可以实现在硅芯片(可姑且理解为"更底层的硬件")上， 从而最大限度地减少了软件和处理器的负担。

1.4 Some Preliminary Terminology | 一些预备术语

- **Processor Node.** See Figure 1-1 on Page 11. A group of one or more processors and their associated main memory. It is interfaced to the IBA fabric via one or more HCAs (Host Channel Adapter), each of which implements one or more IBA ports. 处理器节点。参见第 11 页图 1-1。处理器节点本质上是一个处理器和内存的关联组，其中，处理器的个数大于等于 1。处理器节点通过一个或多个 HCA 连接到 IB Fabric, 每一个 HCA 拥有的端口数大于等于 1。
- **Port.** A bi-directional interface that connects an IBA device to an IBA link. 端口。端口是一个双方向的接口，它负责连接一个 IBA 的设备到 IBA 的物理链路上。
- **Link.** The bi-directional, high-speed connection between two ports on two IBA devices. At a minimum, it is implemented using one high-speed serial transmission line in each direction capable of transmitting at 2.5Gb/s, yielding 250MB/s throughput (note that each 8-bit character is converted to a 10-bit character before transmission). Optionally, a link may be implemented with four or twelve transmission lines in each direction, yielding 1GB/s or 3GB/s throughput, respectively. Link 本质上是一个高速的双方向的链接，连接在两个不同的 IBA 设备之上，连接在它们的端口与端口之间。若提供最小的支持，Link 使用一根高速传输线，该串行传输线支持单方向达到 2.5Gb/s 的传输能力，产生 250MB/s 的吞吐量。(注意：每一个占 8 个比特位的字符在被传输之前会被转换成一个占 10 个比特位的字符，其中增加的两个比特位为校验

码。） 可选地，一个 Link 可以在每一个方向上用 4 根或 12 根串行传输线来实现，每个方向上分别产生 1GB/s 或 3GB/s 的吞吐量。

- **IOU and IOC.** Refer to Figure 1-2 on Page 12. An IO Unit (IOU) is comprised of: IO 单元和 IO 控制器。参见第 12 页图 1-2。一个 IO 单元由如下部件组成：

  – The TCA (**Target Channel Adapter**) that interfaces it to the IBA fabric. TCA(目标通道适配器)。TCA 作为接口将 IO 单元连入 IBA 的 Fabric.

  – One or more IO Controllers (IOCs) providing the interface to various IO devices. An example of an IOC would be a mass storage array. 一个或多个 IO 控制器。IO 控制器作为接口将 IO 单元连接到各种各样的 IO 设备上去。例如：一个大容量的存储阵列就是一个 IO 控制器。
- **CA.** The term Channel Adapter is abbreviated as CA throughout the remainder of this book. 贯穿本书的剩余章节，术语”通道适配器”一律简称为”CA”。

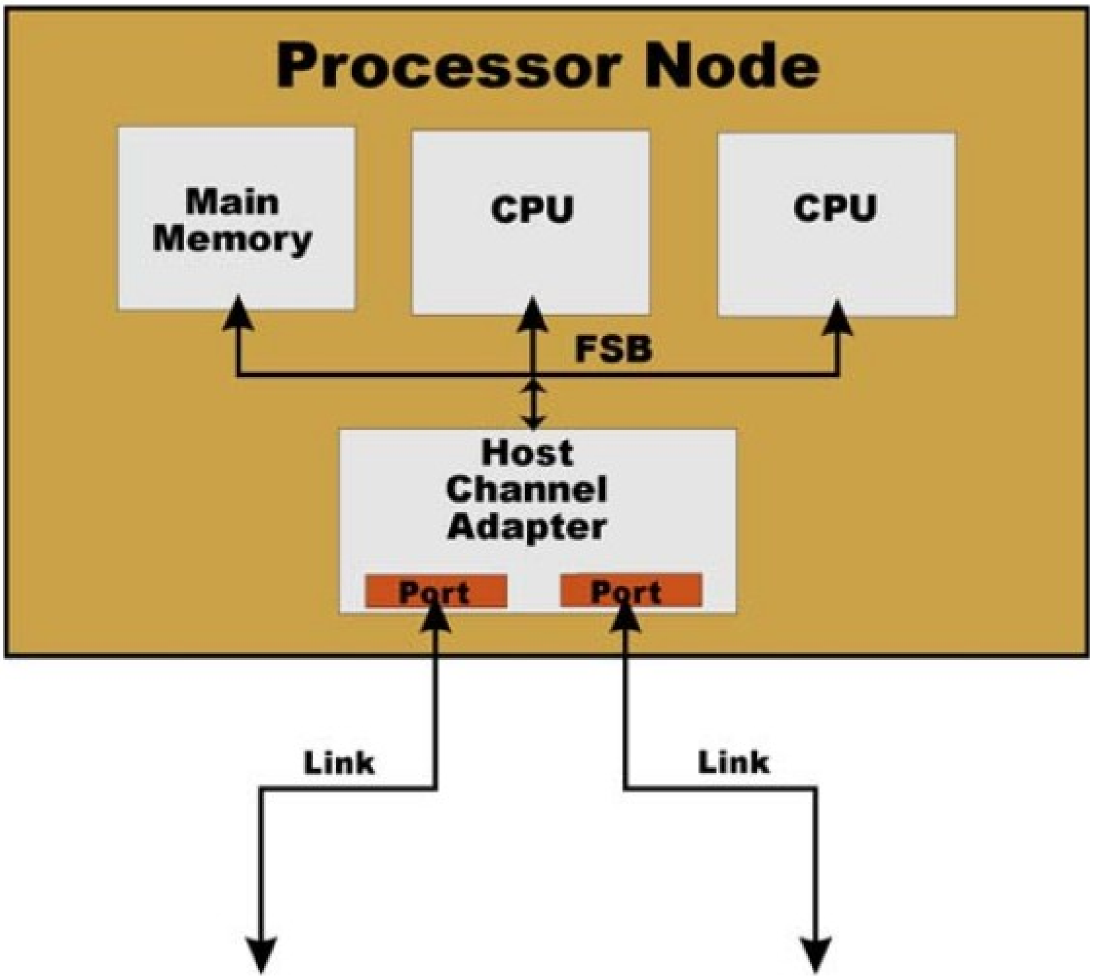

Figure 1-1 : Host Channel Adapter

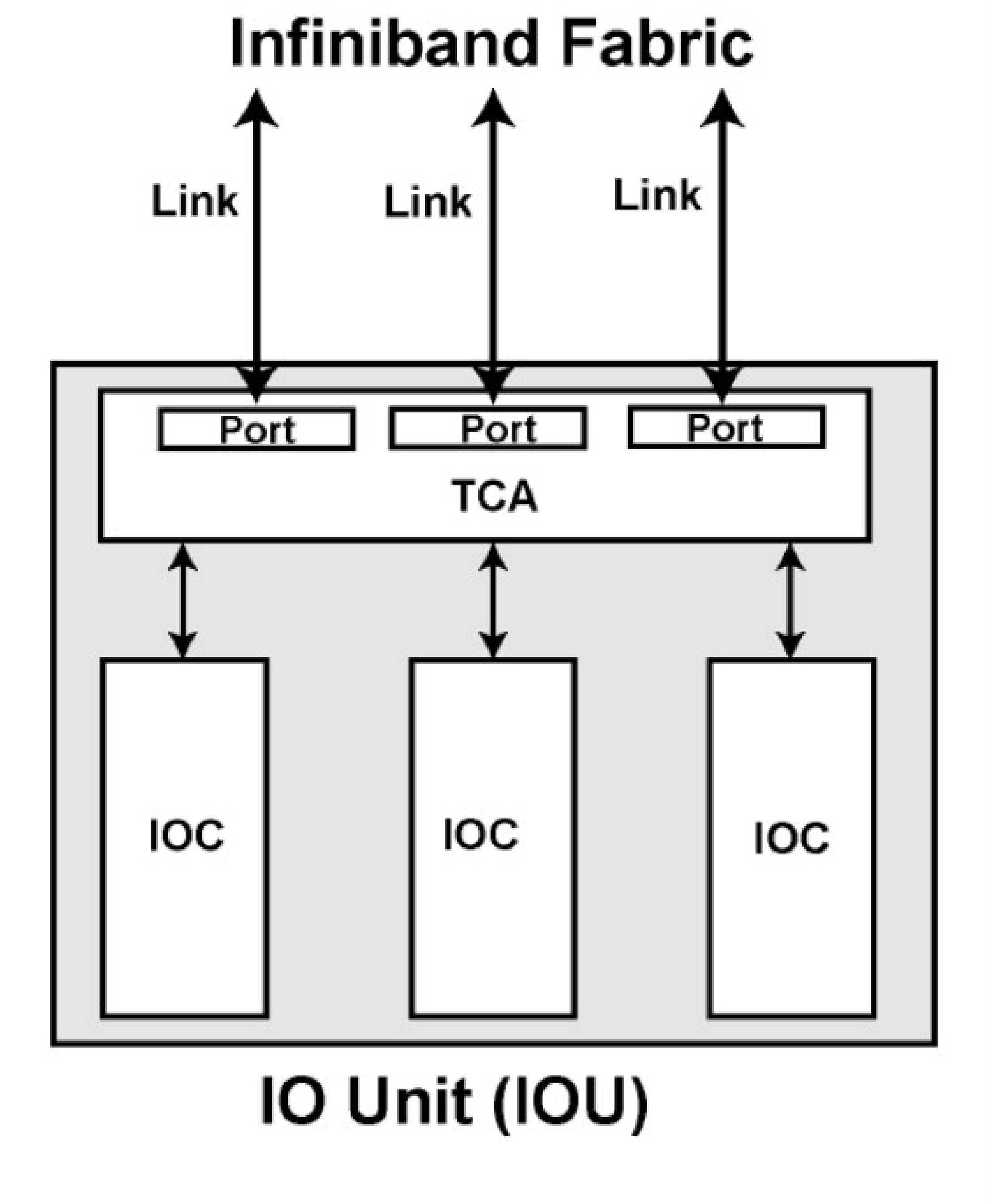

Figure 1-2 : Target Channel Adapter

1.5 Definition Of a Subnet | 子网的定义

The specification defines a subnet as follows: a set of ports and associated links with a common Subnet ID and managed by a common Subnet Manager (SM). 规范对子网的定义如下：子网由一组端口(这组端口共用一个子网 ID 号)和相关联的物理链接构成，一个共同的子网管理器(SM)会对他们进行管理。

The SM is the entity that discovers all of the devices in the subnet at startup time, configures them and then performs a periodic sweep of the subnet to detect any changes to the subnet's topology. 子网管理器(SM)在启动的时候负责发现子网里的所有设备，配置所有设备，然后周期性地扫描整个子网，从而探测到子网拓扑结构发生的任何变化。

Refer to Figure 1-3 on page 13. In the illustration, all of the CA and router ports, as well as switch management ports (switch port 0 – typically an internal port with no connector for a physical link to attach to) that can exchange packets solely by traversing switches (not routers) are said to reside in the same subnet. During configuration, the SM assigns each of these ports a unique Local ID address (referred to as the LID address) as well as a common Subnet ID (also referred to as the Subnet Prefix) that identifies the subnet that a port resides in. Subnets may be connected to each other through routers. 参见第 13 页的图 1-3。在该插图中，所有的 CA 和路由器端口，以及交换机管理端口(注 [1])可以单独地进行消息(报文)交换(消息从源端口出发，抵达目标端口，乃横贯交换机(而非路由器)而过)，于是它们被称为位于同一个子网。子网管理器(SM)在配置过程中，给每一个端口分配一个相同的子网 ID 和一个独一无二的 LID(本地 ID)地址，所有端口共同的子网 ID 也被称为子网前缀，用来标识端口所在的子网。子网与子网之间可以通过路由器相互连通起来。

(注 [1]) 交换机 0 号端口，通常作为内部端口使用。此端口缺乏连接器的支持，因此无法与链路(LINK)相连接。

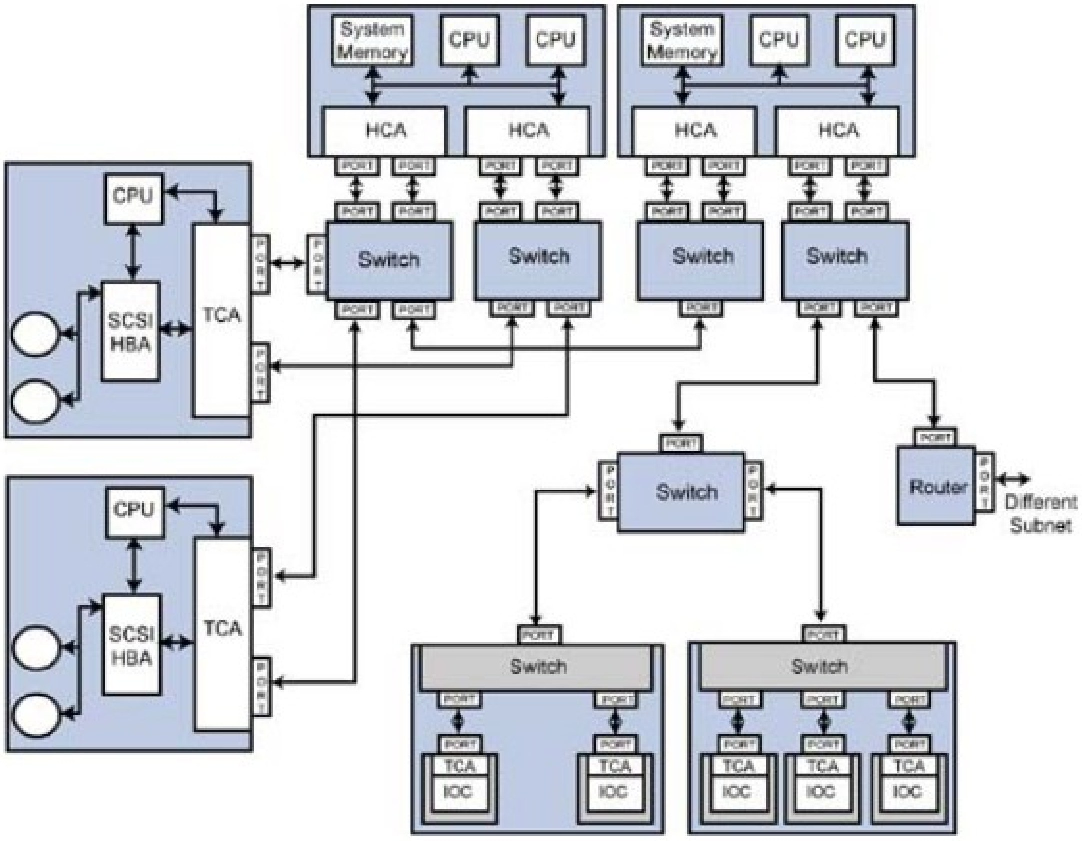

Figure 1-3 : Example System

1.6 Packet Addressing Basics | 分组(包)寻址基础

A packet is used to send a request or a response from one CA to another. One Packet's data payload field can contain a maximum of 4096 bytes of data. An adapter sending a message that is larger than a single packet's data payload field must therefore segment the message transfer into a series of two or more packets. Each packet contains a data payload field, one or two packet routing headers, CRCs, etc. Each packet contains the following address information in its routing headers: 数据包被用来发送一个请求或者响应，从一个 CA 到另一个 CA。在一个包中，数据有效载荷(后面统一译为"干数据")字段最多可以包含 4096 个字节的数据。因此，当适配器发送的消息比单个数据包的干数据字段所能支持的数据上限要大的时候，必须将消息拆分成一系列的(两个或更多)数据包来传送。每一个数据包都包含有干数据字段，一个或两个路由报头，CRC 校验等。在每个数据包的路由报头中，包含着下面的地址信息：

- **Local Route Header(LRH)** is present in every packet (see Figure 1-4 on page 15 and Figure 1-5 on page 15). Among other elements, the LRH contains: 本地路由报头(LRH) 存在于每一个数据包中(见第 15 页的图 1-4 和 1-5)。除其他要素(内容)外，LRH 包括：
  - **Destination port Local ID (DLID).** This 16-bit field is used by switches to guide the packet through the subnet towards the destination port in this subnet. 目标端口 LID(DLID)。DLID 字段占 16 个比特，由交换机使用。交换机用 DLID 来指引数据包穿透子网到达目标端口，该目标端口位于子网的内部。
  - **Source port Local ID (SLID).** This is the 16-bit address of the source port that originates a request packet in a subnet. 源端口 LID(SLID)。SLID 字段占 16 个比特，标识着在子网中发起数据请求包的源端口。
  - Depending on the version of protocol used to perform the transfer, the destination CA may respond to a request packet by returning a response packet. The SLID and DLID fields obtained from the request packet are reversed in the response packet's LRH. 根据用于执行传输的协议版本，目标 CA 通过返回一个响应数据包，对请求数据包做出响应。在响应包的 LRH 中，SLID 和 DLID 字段正好与请求包中的 SLID 及 DLID 字段相反。也就是说，响应包的 SLID 对应于请求包的 DLID, 请求包的 SLID 对应于响应包的 DLID。

- **Global Router Header (GRH).** See Figure 1-5 on page 15. The GRH is only present if the source and destination CAs are not in the same subnet. Among other elements, the GRH contains: 全局路由报头(GRH)。只有当源 CA 与目标 CA 不在同一个子网的时候，GRH 才会存在。除其他要素(内容)外，GRH 包括：
  - **Destination port Global ID (DGID).** This field is used by routers to guide the packet through the fabric towards the subnet within which the destination port resides. It is a 128-bit value. The upper 64-bits identifies the subnet within which the destination CA port resides. The lower 64-bits is referred to as the Globally Unique ID (GUID) and uniquely identifies the destination port. 目标端口 GID(DGID)。这一字段由路由器使用。路由器用 DGID 来指引数据包穿透 fabric 到达目标端口所在的子网。DGID 占 128 个比特，其中，高 64 位用来标识目标端口所在的子网，低 64 位被称为

GUID，其能够唯一地标识目标端口。

- **Source port Global ID (SGID).** This is the 128-bit address of the source port that originates a request packet. The upper 64-bits identifies the subnet within which the source CA port resides. The lower 64-bits is the source port's GUID that uniquely identifies the source port. 源端口 GID(SGID)。SGID 是一个 128 位的地址，标识着发送数据请求包的源端口。其中， 高 64 位用来标识源端口所在的子网，低 64 位是源端口的 GUID，其能够唯一地标识源端口。
- Depending on the version of the protocol used to perform the transfer, the destination CA may respond to a request packet by returning a response packet. The SGID and DGID fields obtained from the request packet are reversed in the response packet's GRH. 根据用于执行传输的协议版本，目标 CA 通过返回一个响应数据包，对请求数据包做出响应。在响应包的 GRH 中，SGID 和 DGID 字段正好与请求包中的 SGID 及 DGID 字段相反。也就是说，响应包的 SGID 对应于请求包的 DGID, 请求包的 SGID 对应于响应包的 DGID。

LRH	BTH	Other headers	Packet Data	CRCs

Figure 1-4. Packet Format When the Source and Destination Ports Are In the Same Subnet

LRH	GRH	BTH	Other headers	Packet Data	CRCs

Figure 1-5. Packet Format When the Source and Destination Ports Are In Different Subnets

1.7 Every Packet Contains a BTH

基本传输报头(BTH)存在于任何一个数据包之中

Refer to Figure 1-4 on this page and Figure 1-5 on this page. In addition to the LRH and, possibly, the GRH, every packet contains a Base Transport Header (BTH). Among other things, the BTH contains: 参见本页的图 1-4 和 1-5。除了必须存在的 LRH 和可能存在的 GRH 之外，每个数据包都包含有一个基本传输报头(BTH)。除其他事项外，BTH 还包括:

- The Opcode field identifies the type of request or response packet. 操作码字段，标识请求或响应数据包的类型。
- The DestQP field identifies the destination QP within the remote CA. The destination QP is responsible for handling the incoming packet. QPs are described in the chapter entitled, “QP: Message Transfer Mechanism” on page 31. 目标 QP 字段，标识远端 CA 内的 QP。目标 QP 负责处理传入的数据包。关于 QP 的描述，请参见第 31 页的标题为”QP：消息传输机制”之章节。
- The PSN field contains the packet's 24-bit Packet Sequence Number. The PSN is inserted into a transfer request packet by the initiator and is checked for correctness by the recipient. This permits the recipient to detect missing packets. PSN 字段，包含着 24 位的数据包序列号。发件方(发起人)将 PSN 插入到传输请求数据包，收件方(收件人)在收到数据包后检查其正确性。这样，丢包

就能够被收件方所检测到。

1.8 Channel Adapters | 通道适配器(CA)

Sometimes the specification uses the term xCA when it referring to sometihng that is applicable to either a TCA or HCA. IB 规范有时候会使用术语 xCA 来指代 TCA 或者 HCA。

1.8.1 CAs Are the Real Players (Switches and Routers Are Just Traffic Cops) | CA 是真正的驾驶员(车手)，交换机和路由器只是交通警察而已

The Real players are the CAs. When a CA must send information to or read information from another CA, the request to do so is output through one of the CA's ports in the form of a request packet. Each CA port has a unique address assigned to it during configuration. The request packet contains the address of the destination port on the destination CA. 真正的参赛选手是 CA。当一个 CA 必须给另一个 CA 发送信息或者必须从另一个 CA 那里读取信息的时候，收发信息的请求将以请求数据包的形式，穿过 CA 的端口径直输出出去。SM 在配置过程中给每一个 CA 分配了一个唯一的地址。请求数据包包含了目标 CA 的目标端口地址。

Once the request packet is transmitted into the fabric by a CA, it is guided through the subnet by switches. If the destination CA port is in a different subnet, the packet must also transit one or more routers to get to it. 一旦 CA 将请求数据包发送到 fabric 中去以后，请求包就在交换机的指引下穿越子网直至到达目的地。如果目标 CA 端口与源 CA 端口不在同一个子网，(除了被交换机转发外)，请求包也必须经由一个或多个路由器的转发方能到达目的地(即目标 CA 之目标端口)。

1.8.2 Endnode = CA | 终端节点 = CA

An endnode is defined as a device other than a switch, router or a repeater: in other words, it's a CA. The endnode acts either as the initiator or the ultimate recipient of a packet. The specification's exact definition of an endnode is: 终端节点被定义为除了交换机，路由器和中继器之外的设备。换言之，终端节点就是 CA(通道适配器)。CA 要么担任数据包的发件方，要么充当数据包的收件方。IB 规范对终端节点的确切定义如下：

> “An endnode is any node that contains a Channel Adapter and thus it has multiple queue pairs and is permitted to establish connections, end to end context, and generate messages. Also referred to as Host Channel Adapter or Target Channel Adapter, two specific types of endnodes.” “终端节点是包含有通道适配器的任何节点。因此，终端节点拥有多个 QP，并且被允许用来建立连接及端到端的上下文，和生成消息。终端节点也被称为 HCA 或 TCA,它们是两种特定类型的终端节点。”

1.9 Role of Switches and Routers | 交换机和路由器的作用

When a request packet is issued by a CA, there are two possibilities: 当 CA 发出一个请

求数据包的时候，有如下两种可能性：

1. Source and destination CAs directly connected. Refer to Figure 1-6 on page 17. The destination CA may be directly connected to the CA issuing the request packet. In this simple case, the request packet is transmitted over a single link and is received by the destination CA port. The port decodes the packet's DLID address field, determines that it is the targeted port, accepts the request packet, and processes the request. 源 CA 与目标 CA 直接相连。参见第 17 页图 1-6。在这种简单情况下，请求数据包在单一链路上传输，目标 CA 端口能径直接收到请求数据包。目标端口首先解码接收到的数据包中的 DLID 地址字段，然后确定请求包的目标端口正是它自己，于是接受并处理该请求。

2. Source and destination CAs aren't directly connected. Refer to Figure 1-3 on page 13. The destination CA may not be directly connected to the CA issuing the request packet. In this case, when the request packet is output onto the first link by the CA issuing the request, it arrives at a port on a switch or a router: 源 CA 与目标 CA 并不直接相连。参见第 13 页图 1-3。目标 CA 可能与发出请求数据包的源 CA 不直接相连。在这种情况下，当发出请求的 CA 把请求数据包输出到第一个链路上的时候，请求数据包到达交换机或者路由器的端口上：

- **Switch's Role.** Refer to Figure 1-3 on page 13. Switches route packets within a local network referred to as a subnet. Using the packet's destination port address field (DLID), the switch performs a lookup in its Forwarding Table to determine through which of the switch's ports the packet must be transmitted to move it towards the destination CA port. The switch's Forwarding Table is set up by the configuration software (i.e., the SM) at startup time. The packet may have to transit one or more switches before it arrives at the destination CA port. 交换机的作用。参见第 13 页图 1-3。交换机为数据包在子网内部(即本地网络)实施路由查找。交换机在其转发表中查找数据包中的目标端口地址字段(DLID)，从而确定必须把数据包发送到哪一个交换机端口上。那个交换机端口会将数据包朝着目标 CA 端口的方向移动。交换机的转发表是配置软件 SM 在其启动的时候就设置好了的。
- **Router's Role.** Refer to Figure 1-3 on page 13. When the source and destination CAs reside in different subnets, the request packet contains a GRH (in addition to the LRH). As switches within a subnet move the request packet towards the destination port, the packet eventually arrives at a router port. The router uses the request packet's GRH:DGID port address to determine the subnet in which the destination CA resides. Specifically, it uses the packet's destination subnet ID field (i.e., the upper 64 bits of the 128-bit DGID address) to perform a lookup in its Routing Table, and then re-transmits the packet through the port indicated by the selected table entry. The router's Routing Table is set up by the configuration software at startup time. The packet may have to transit one or more routers before it arrives at the destination CA port. 路由器的作用。参见第 13 页图 1-3。当源 CA 与目标 CA 不在同一个子网的时候，请求数据包中除了包含 LRH 外，还包含有 GRH。当与源 CA 同处一个子网的交换机将请求数据包朝着目标端口

方向移动的时候，请求数据包最终会抵达路由器的端口上。路由器通过请求包中的 GRH:DGID 来确定目标 CA 所处的子网。具体而言，路由器在其路由表中查找数据包中的目标子网 ID 字段(即 128 位的 DGID 地址的高 64 位)，然后根据查找到的路由表条目的指示再次转发数据包。路由器的路由表是路由器配置软件再启动时候就设置好了的。数据包到达目标 CA 端口之前，可能不得不经停(穿越)一个或多个路由器。

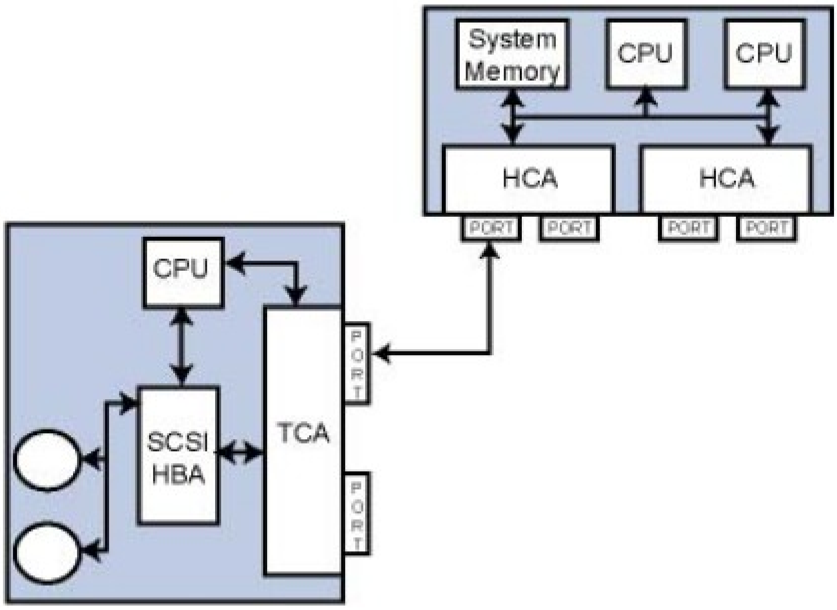

Figure 1-6. Sample System Block Diagram

1.10 Repeater's Role | 中继器的作用

In order to compensate for weakened signal strength and/or the build up of jitter between the two ends of a long link run, a retiming repeater (sometimes just referred to as a repeater) may be placed in the link. 为了补偿已经衰减的信号强度，和(或)增强位于一条长链路两端之间的信号振动，一个重定时中继器(有时只称为中继器)可能被放置在该物理链路(LINK)中。

1.11 It's All About Message Passing | 一切的一切，所有的一切都是关于消息传递

The underlying concept behind IBA is message passing between CAs. A message is a block of data that is passed from the local memory of one CA to the local memory of another CA. 从本质上讲，隐藏在 IBA 背后的概念就是 CA 之间的消息传递。(什么是消息？) 消息就是数据块。 (什么样的数据块？) 从一个 CA 的卡内存被传递到另一个 CA 的卡内存的数据块。

1.11.1 Specification Usage of "Message" and "Packet" | IB 规范中关于”消息”和”包”的用法

The reader should note that the specification doesn't always use the terms "message" and "packet" correctly. Sometimes "message" is used where "packet" would be correct and vice versa. 读者朋友请注意，(我们的至上权威)IB 规范对术语"消息"和"包"的使用并不总是正确的。有时候，使用"消息"的地方，其实用"包"才对；反之亦然。

1.11.2 Three Types of Message Transfers | 消息传输的三种类型

There are three basic message transfer scenarios: 有三种基本的消息传递场景如下所示:

- Sending a message from the local CA's memory to the destination CA's memory. 发送消息，将消息从本地 CA 卡内存发送到目标 CA 卡内存里去。
- Reading a message from the destination CA's memory and storing it in the local CA's memory. 读取消息，从目标 CA 卡内存里去读消息，存入本地 CA 的卡内存。
- Performing an atomic RMW (read/modify/write) in the destination CA's memory and storing the returned data in the local CA's memory. 原子 RMW 操作，在目标 CA 的卡内存中执行原子的读取/修改/写入操作，将返回的数据存入本地 CA 的卡内存。

1.11.2.1 Writing a Message to the Remote CA's Memory | 将消息写入远端 CA 内存

In this case, a CA sends a message from its own local memory to the destination CA's local memory. There are two scenarios: 在这一个例中，CA 将一条消息从它自己的卡内存发送到目标 CA 的卡内存中去，分为两种场景：

- **Message Send Operation.** In this case, the request doesn't tell the destination CA where to write the data in its local memory. Rather, it's up to the destination CA to determine where to write the data in its local memory. In IBA, this is referred to as a Send operation. 消息发送操作。在这种情况下，请求并不告知目标 CA 从什么地方将数据写入内存。相反，目标 CA 自行确定从什么地方写数据。在 IBA 中，这种操作被称为 SEND 操作。

- **Remote DMA (RDMA) Write Operation.** In this case, the request specifies where the data is to be written in the destination CA's local memory. In addition to the write data contained in the data payload field, the request packet contains: RDMA 写操作。在这种情况下，请求指定从什么位置将数据写入目标 CA 卡内存。除了包含在干数据字段中的待写入数据外，请求数据包还包括:
  - the memory start address, 内存起始地址，
  - the transfer length, and 传输长度，和
  - a special key indicating that it has permission to perform the write. 一个特殊的键(钥匙)，此 key 表明源 CA 拥有执行写操作的权限。

  In IBA, this is referred to as an RDMA Write operation. 在 IBA 中，这被称之为 RDMA 写操作。

1.11.2.2 Reading a Message From the Remote CA's Memory

从远端 CA 卡内存中读消息

In IBA, this is referred to as a Remote DMA Read (RDMA Read) operation. A CA issues a request to another CA to read a block of requested read data from its local memory and return it to the requester in a series of one or more RDMA Read response packets. Upon receipt of the requested read data, the requesting CA stores it in a specified area of its own local memory. The RDMA Read request packet contains: 在 IBA 中，这被称之为 RDMA 读操作。一个 CA 向另一个 CA 发出请求，请求读取数据块，被请求的 CA 返回给请求方一系列(一个或多个)的 RDMA 响应包。请求的 CA 在接收到请求读取的数据之后，将数据存入其卡内存的特定区域。RDMA 读请求包包括:

- The memory start address. 内存的起始地址。
- The amount of data to be read. 待读取的数据量。
- A special key indicating that it has permission to read data from that area of the destination CA's local memory. 一个特殊的键(钥匙)，此 key 表明源 CA 拥有从目标 CA 的(特定)内存区域中读取数据的权限。

1.11.2.3 Performing an Atomic RMW in the Remote CA's Memory

在远端 CA 内存中执行原子 RMW 操作

In this case, a CA wishes to perform an atomic RMW (read/modify/write) in the destination CA's local memory. In IBA, there are two forms of atomic RMW operations: 在这种情况下，CA 希望在目标 CA 的卡内存中执行原子的 RMW(读取/修改/写入)操作。在 IBA 中，原子的 RMW 操作分为两种形式:

1. Atomic Fetch and Add operation. The CA issuing the request supplies the destination CA with: 原子提取和添加操作(注 1)。发起请求的 CA，给目标 CA 提供:

- the memory address, 内存地址，
- an Add value, 待添加的(数)值，
- and a special key indicating that it has permission to access the location in the destination CA's local memory. 和一个特殊的键(钥匙)，此 key 表明源 CA 拥有访问目标 CA 内存的(特定)位置的权限。

  (注 1) 原子提取与操作(fetch-and-add)，通常被用来实现并发控制结构，例如互斥锁和信号量。更多介绍请参见网页 https://en.wikipedia.org/wiki/Fetch-and-add

Upon receipt of the request, the destination CA reads from the target location in its local memory, adds the Add value to the value read, and writes the result back into the local memory location. The destination CA returns the initial value read back to the CA that issued the request in an Atomic Response packet. Upon receipt of the response packet, the requesting CA writes the read data into its own local memory. 目标 CA 在收到请求后，从它的局部内存中按照请求包中指定的目标位置 (A)去读取数据，将待加的数值(Y)与读取到的数值(X = [A])相加，然后把相加的结果(X+Y)写回原地([A]=X+Y)。目标 CA 返回给发出请求的 CA 一个原子响应包，其中包含了目标 CA 读取到的原始数值(X)。发出请求的 CA 将目标 CA 读取到的数据(X)写入它自己的本地内存。

2. Atomic Compare and Swap If Equal operation. The CA issuing the request supplies the destination CA with: 原子比较与交换操作(注 2)。发起请求的 CA，给目标 CA 提供:

- the memory address, 内存地址，
- a Compare value, 待比较的(数)值，
- a Swap value, 待交换的(数)值，
- and a special key indicating that it has permission to access the location in the destination CA's local memory. 和一个特殊的键(钥匙)，此 key 表明源 CA 拥有访问目标 CA 内存的(特定)位置的权限。

  (注 2) 原子比较与交换操作(CAS)，通常被用在多线程中而实现同步。更多介绍请参见网页 https://ien.wikipedia.org/wik/Compare-and-swap

Upon receipt of the request, the destination CA reads from the target location in its local memory, compares the data read to the Compare value, and, if they are equal, writes the Swap value into the location. The destination CA returns the initial value read to the CA that issued the request in an Atomic Response packet. Upon receipt of the response packet, the requesting CA writes the read data into its own local memory. 目标 CA 在收到请求后，从它的卡内存中按照请求包中指定的目标位置(A)去读取数据，将读取到的数据(X)与待比较的数值(C)进行比较，如相等(X==C)，则将待交换的数值 (S)写回原地([A]=S)。目标 CA 返回给发出请求的 CA 一个原子响应包，其中包含了目标 CA 读取到的原始数值(X)。发出请求的 CA 将目标 CA 读取到的数据(X)写入它自己的卡内存。

1.11.3 What's in a Message? | 消息里究竟有什么？

How the recipient of a message interprets the message is device-specific. 消息收件方如何翻译(诠释)消息，因设备而异，和设备本身密切相关。

1.11.3.1 Example Disk Read Request | 案例：读取磁盘请求

Step One: Disk Read Issued Via a Message Send Operation
第一步：通过发起一个消息 **SEND** 操作去读磁盘

For example, a message might be passed to a mass storage controller using a message Send operation. In this example, the message may contain: 例如，通过消息 SEND 操作，一条消息可能被传递给一个大容量存储器。在这个例子中，消息可能包括如下内容:

- The type of operation to be performed (disk read or write). 将要执行的操作(磁盘读或写)类型。
- The identity of the target disk drive (if the disk controller controls an array of disk drives). 目标磁盘驱动器的身份(ID)(如果磁盘控制器掌控着一个磁盘阵列)。
- The start cylinder number. 起始柱面号。
- The surface number. 表面号。
- The start sector number. 起始扇区号。
- The number of sectors to be read or written. 要读取或写入的扇区个数。
- If it's a write operation, the data to be written to disk. 若为写操作，待写入磁盘的数据。

- If it's a read operation, where the return data is to be written in the requesting CA's local memory, as well as a special key that will indicate the disk controller has permission to write to that area of the requesting CA's local memory. 若为读操作，即将返回的数据应该写入到发起请求的 CA 的本地内存位置，和一个特殊的键(钥匙)，该 key 表明磁盘控制器对发起请求的 CA 的本地内存拥有写权限。

Step Two: Data Read From Disk
第二步：从磁盘读取数据

Continuing with this example, after receiving the message containing the above information (sent via the message Send operation), the disk controller determines that it's a disk read request. It reads the requested data from the target drive and stores it in its local memory. 继续这个例子, 磁盘控制器在接收到包含上述信息的消息后，确定其为一条读磁盘的请求，于是磁盘控制器去目标磁盘驱动器上读取请求的数据，然后将读取到的数据存入磁盘控制器的本地内存。

Step Three: Read Data Sent Back Via RDMA Write
第三步：通过 **RDMA** 写操作将读取的数据发送回去

Upon completing the disk read, the disk controller then initiates an RDMA Write operation to write the requested disk data into the area of the requesting CA's local memory that was identified in the original message. In the RDMA Write request packet, it supplies the requested data in the packet's data payload field, and also supplies the memory start address, the transfer length, and the special key that it received in the original message. 磁盘控制器在完成磁盘读操作后，启动 RDMA Write(写)操作，将请求的读取的硬盘数据写入到请求 CA 的本地内存区域，此内存区域的身份在最初的消息中已经被识别。在这个 RDMA Write(写)请求包中，提供了请求读取的数据(被填入数据有效载荷字段)，也提供了(发出请求的 CA 的)内存的起始地址，传输的长度，和从最初的消息中收到的特殊的 Key(密钥)。

Step Four: Upon Receipt of RDMA Write Request
第四步：源 **CA** 在收到 **RDMA** 写请求后

Upon receipt of the RDMA Write request packet, the CA that originated the disk read request checks the special key to ensure the writer has permission to write to the indicated area of its local memory. Assuming that the key is correct, it then writes the data into its local memory. Upon completion of the write, the CA signals completion to software (perhaps via an interrupt). 发出读取磁盘数据请求的 CA(即源 CA)在收到 RDMA Write(写)请求后，首先检查那个特别的 Key(即密钥)，确保写的一方有权限将数据写入到它的局部内存里的指定区域里去。当写操作完成后，源 CA 发信号(可能通过中断)通知软件，任务已完成。

1.11.4 How Big Can a Message Be? | 一条消息能有多大?

An IBA message transfer can be anywhere from zero to 2GB in size. IBA 规定，一条消息的长度可以介于 0 字节到 2G 字节之间。(简言之，0-2GB)

1.11.5 What Is the Maximum Size of a Packet's Data Payload Field? | 数据包的干数据字段的最大尺寸是多少？

An IBA packet can contain a maximum of 4KB of data. For additional information, refer to "Maximum Data Payload Size: on page 42. IBA 规定，一个数据包可以容纳最大为 4K 字节的数据。关于其详细信息，请参见第 42 页”最大的干数据尺寸”。

1.11.6 Large Messages Require Multiple Packet Transfers | 传输大消息，需要拆分成多个包发送

It should be obvious that when a CA wishes to transfer a message that exceeds the size of a packet's data payload field, the CA must perform a multiple packet transfer in order to transfer the entire message. The following sections define the characteristics of multiple packet message transfers for the various message transfer operation types. It should be noted that in some circumstances, a packet's data payload field may be constrained to a size smaller than 4KB. For more information, refer to “Maximum Data Payload Size” on page 42. The following subsections assume that the maximum allowable data payload field size is 4KB. 显而易见的是，当 CA 希望传输一条消息(消息的尺寸已经超过了干数据字段的大小)，CA 必须执行多个分组传送，才能够将整条消息发送出去。以下各节定义了各种消息传输类型的多个分组消息传输的特性。应该指出的是，在某些情况下，一个包中的干数据字段可能被限制在比 4K 还要小的某一尺寸。更多信息，请参见第 42 页的”干数据的最大尺寸”。下面的各小节中，假定允许的干数据尺寸为 4KB。

1.11.6.1 Each Packet Contains an Opcode Field | 每个包都包含操作码字段

Each packet contains an Opcode field that defines the type of request or response packet. The sections that follow provide some detail on the Opcode types. 每一个包中都包含了一个操作码字段，该操作码定义了请求包或者响应包的类型。接下来的部分提供了操作码类型的一些细节。

1.11.6.2 Some Request Types Require a Response While Others Don't | 有些请求类型需要响应，有些则不需要

While the destination CA is required to return a response for an RDMA Read request (the requested read data is returned) or for either type of atomic RMW request (the data read from the destination CA's local memory is returned), no response is required to be returned for a Send or an RDMA Write operation (although some of the service types require the return of an Acknowledge packet; this is discussed in a later chapter). 目标 CA 需要对 RDMA 读请求做出响应(返回请求读的数据)或对任意一种类型的原子 RMW 请求做出响应(从目标 CA 卡内存中读到的数据，返回给请求

CA)；目标 CA 不需要对 SEND 或者 RDMA 写操作做出响应(虽然某些服务类型需要返回一个应答(ACK)包，这在后面的章节将给予展开讨论)。

1.11.6.3 Single- and Multi-Packet Send Operations | 单包和多包 SEND 操作

1.11.6.3.1 Single Packet Send | 单包 SEND

When the message to be sent is no more than 4KB in size, a single request packet with a "Send Only" opcode and a data payload field containing somewhere between zero and 4KB of data is sent to the destination CA. 当待发送的消息不超过 4KB 时，单个请求包被发送给目标 CA。该单请求包的操作码为"只发送"(Send Only)，干数据字段包含的数据长度在 0 到 4KB 之间。

1.11.6.3.2 Multiple-Packet Send | 多包 SEND

However, when the size of the message to be sent exceeds the size of a packet's data payload field, the Send operation consists of a series of two or more Send request packets: 然而，当待发送的消息长度超过 4KB 时，SEND 操作由一系列的(2 个或更多)SEND 请求包组成：

- When the message to be sent is more than 4KB but not greater than 8KB in size, two request packets are sent to the destination CA: 当待发送的消息长度(记为 N)超过 4KB 但是不超过 8KB 时(4KB<N<=8KB)，消息被分拆成两个请求包发送给目标 CA：
  - a request packet with a "Send First" opcode with a data payload field containing 4KB of data, 第一请求包，操作码为"Send First"，干数据字段包含 4KB 的数据，
  - followed by a request packet with a "Send Last" opcode containing somewhere between one byte and 4KB of data. 接下来的请求包(即第二个请求包)，操作码为"Send Last"，1B<=干数据长度<=4KB。

- When the message to be sent is more than 8KB in size, three or more request packets are sent to the destination CA: 当待发送的消息长度(记为 N)超过 8KB 时(N>8KB)，消息被分拆成多个包(三个或更多)发送给目标 CA：
  - a request packet with a "Send First" opcode with a data payload field containing 4KB of data, 第一请求包，操作码为"Send First"，干数据字段包含 4KB 的数据，
  - followed by one or more request packets, each with a "Send Middle" opcode and 4KB of data, 接下来的多个包(至少一个)，操作码为"Send Middle"，干数据字段包含 4KB 的数据，
  - followed by a request packet with a "Send Last" opcode containing somewhere between one byte and 4KB of data. 最后一个包，操作码为"Send Last"，1B<=干数据长度<=4KB。

1.11.6.4 Single- and Multi-Packet RDMA Write Operations | 单包和多包 RDMA 写操作

1.11.6.4.1 Single Packet RDMA Write | 单包 RDMA 写

When the size of the message to be written is such that the entire message fits in a single request packet's data payload field, a single request packet is transmitted with an "RDMA Write Only" opcode and a data payload field containing somewhere between zero and 4KB of data. 当待写入消息正好充满干数据字段，单个请求包将被发送，操作码为”RDMA Write Only”，0<=干数据长度<=4KB。

1.11.6.4.2 Multiple Packet RDMA Write | 多包 RDMA 写

When the size of the message to be written exceeds the size of a packet's data payload field, the RDMA Write operation consists of a series of two or more RDMA Write request packets. There are two possible scenarios: 当待写入数据长度超过干数据字段支持的最大尺寸(4KB)时，RDMA 写操作由一系列的(2 个或更多)RDMA 写请求包构成。由两种可能的场景：

- A request packet with an "RDMA Write First" opcode with a data payload field containing 4KB of data, followed by a packet with an "RDMA Write Last" opcode containing somewhere between one byte and 4KB of data. 第一个包，操作码为"RDMA Write First"，干数据长度为 4KB；第二个包，操作码为"RDMA Write Last"， 1B<=干数据长度<=4KB。

- A request packet with an "RDMA Write First" opcode with a data payload field containing 4KB of data, followed by one or more request packets, each with an "RDMA Write Middle" opcode and 4KB of data, followed by a packet with an "RDMA Write Last" opcode containing somewhere between one byte and 4KB of data. 第一个包，操作码为"RDMA Write First"，干数据长度为 4KB；接下来的包(一个或多个)，操作码为"RDMA Write Middle"，干数据长度为 4KB；最后一个包，操作码为"RDMA Write Last",1B<=干数据长度<=4KB。

It should be noted that the first or only request packet of an RDMA Write operation, in addition to a data payload field, also contains the start memory address, the amount of data to be written, and the special key indicating that it has permission to write the data to the destination CA's local memory. 应该指出的是，在 RDMA 写操作中，第一个(或唯一的)请求包中，除了包含干数据字段外，还包含内存的起始地址，待写入的数据量，和特殊的键(密钥)，该键表明它拥有将数据写入目标 CA 卡内存的权限。

1.11.6.5 RDMA Read Operation | RDMA 读操作

It only takes one request packet to issue an RDMA Read request to a destination CA. The destination CA then returns the requested read data to the requesting CA in a series of one or more RDMA Read response packets. 对目标 CA 发起一个 RDMA 读请求只需要一个请求包，目标 CA 随后给源 CA 返回读取到的数据，通过一系列的(一个或多个)RDMA 读相应包。

- If all of the requested read data fits into a single packet, then a single RDMA Read response packet is returned with an opcode of "RDMA Read Response Only". The packet's data payload contains between zero and 4KB of data. 如果请求读取到的数据可以填入单个包，单个 RDMA 读响应包将被返回。操作码为"RDMA Read Response Only"，0<=干数据长度<=4KB。

- If all of the requested read data will fit into two response packets, then: 若请求读取到的数据将填入两个包：
  - an "RDMA Read Response First" packet with a data payload containing 4KB of data is returned, 返回的第一包，操作码为"RDMA Read Response First"，干数据长度为 4KB，
  - followed by an "RDMA Read Response Last" packet containing somewhere between one byte and 4KB of data. 返回的第二个包，操作码为"RDMA Read Response Last", 1B<=干数据长度<=4KB。

- If the requested read data requires more than two response packets, then: 若请求读取到的数据将填入多个包(两个以上)：
  - an "RDMA Read Response First" packet with a data payload containing 4KB of data is returned, 返回的第一包，操作码为"RDMA Read Response First"，干数据长度为 4KB，
  - followed by one or more "RDMA Read Response Middle" packet -s, each with a data payload of 4KB, 接下来返回的一个或多个包，操作码为"RDMA Read Response Middle"，干数据长度为 4KB，
  - and, finally, an "RDMA Read Response Last" packet containing somewhere between one byte and 4KB of data. 返回的最后一个包，操作码为"RDMA Read Response Last"，1B<=干数据长度<=4KB。

1.11.6.6 Atomic Operation | 原子操作

It only takes one request packet to issue the request and one response packet to return the read data. 发起请求只需要一个请求包，目标 CA 做出响应也只要返回一个响应包。

- The request packet contains either a "CmpSwap" or "FetchAdd" opcode, as well as the Compare and Swap data, or the Add data. 请求包中包括(1)操作码为"CmpSwap"，待比较和交换的数据，或(2)操作码为"FetchAdd"，待加的数据。

- The single response packet contains an "Atomic Acknowledge" opcode, as well as the data read initially from the targeted memory location in the destination CA's local memory. 响应包中包括操作码为"Atomic Acknowledge"(原子的 ACK)，从目标 CA 卡内存中的目标位置读取到的原始数据。