目录

[第一章 InfiniBand、iWARP和 RoCE 3](#_Toc139471737)

[1.1 简介 3](#_Toc139471738)

[1.2 InfiniBand架构 3](#_Toc139471739)

[1.3 IB 组网 4](#_Toc139471740)

[1.3.1 网络拓扑 4](#_Toc139471741)

[1.3.2 网络拓扑 5](#_Toc139471742)

[1.4 通信机制 7](#_Toc139471743)

[1.4.1 通信服务(传输) 7](#_Toc139471744)

[1.4.2 寻址 8](#_Toc139471745)

[1.4.3 数据包格式 9](#_Toc139471746)

[1.5 分层体系结构 10](#_Toc139471747)

[1.5.1 物理层 11](#_Toc139471748)

[1.5.2 链路层 12](#_Toc139471749)

[1.5.3 网络层 15](#_Toc139471750)

[1.5.4 网络层 15](#_Toc139471751)

# InfiniBand、iWARP和 RoCE

## 简介

InfiniBand (IB) 是一种点对点互连。其零复制和远程直接内存访问  (RDMA) 等功能可直接将数据从发送方内存传输到接收方内存，而无需涉及主机处理器，从而有助于减少处理器开销。本章涵盖整体  IB 架构 (IBA) 及其各个层。本章的重点是链路层和网络层。随着  IB  不断发展以通过以太网为低延迟应⽤程序提供连接，互联网广域RDMA 协议(iWARP) 和  RoCE 正在成为向应⽤程序提供 RDMA功能的有吸引力的选择。InfiniBand (IB) 是一种点对点互连。其零复制和远程直接内存访问(RDMA)等功能可直接将数据从发送方内存传输到接收方内存，而无需涉及主机处理器，从而有助于减少处理器开销。本章涵盖整体IB 架构 (IBA) 及其各个层。本章的重点是链路层和网络层。随着IB不断发展以通过以太网为低延迟应用程序提供连接，互联网广域 RDMA协议  (iWARP) 和  RoCE 正在成为向应用程序提供RDMA功能的有吸引力的选择。本章将详细介绍这两种协议。

## InfiniBand架构

IBA定义了一种交换通信结构，允许许多设备在受保护的远程管理环境中以高带宽和低延迟并发通信。一个终端节点可以通过多个IBA端口进行通信，并可以利用多条路径通过IBA fabric。图1.1展示了IB[10]系统网络中的各个组成部分。该网络由各种处理器节点和I/O单元组成，通过级联交换机和路由器连接在一起。它允许处理器间通信的低延迟互连，支持存储设备到存储设备的连接，并且还证明了路由器可以用于扩展到广域网(WANs)、局域网(LANs，通过以太网)或存储区域网络(SANs)的连接。路由器还提供多个IB子网之间的连接。

IBA还定义了允许与其他第2层技术(如小型计算机系统接口(SCSI)、以太网和光纤通道(FC))通信的体系结构组件。

IBA定义了一个分层协议，它指定了物理层、链路层、网络层、传输层和上层。它定义了各种媒体上的通信，包括印刷电路板(PCBs)、铜和光纤电缆。IB允许三种链路速度: 超过4线(1X:单线)，16线(4X: 4线)，或48线(12X: 12线)。如果单通道(4线:差分对RX+/RX-和TX+/TX-)在2.5 Gbps，然后4X连接提供10Gbps, 12X连接提供30Gbps的通信。

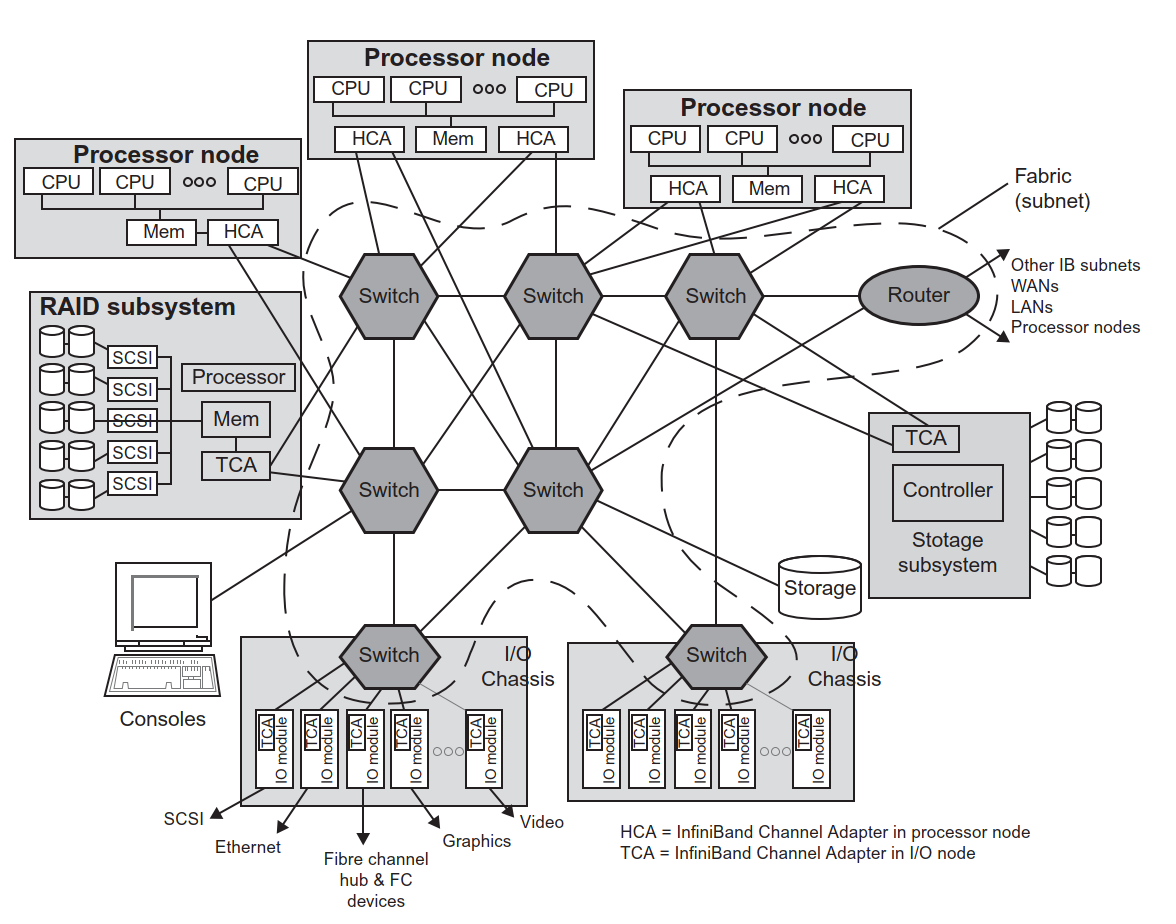


图 1.1 IB结构和组成

IBA支持节点间的单播和组播流量。这种流量可以在可靠或不可靠的模式下进行。它还支持连接或数据报模式进行通信。该体系结构中提供了各种QoS(服务质量)机制，以保证网络中的无损和差异化流量。下面几节将更详细地描述IBA的所有这些方面。

## IB 组网

### 网络拓扑

IBA网络由多个子网组成，子网之间可以通过路由器相互连接。每个终端节点可以是一个处理节点、一个I/O单元或一个存储子系统。IBA允许使用RDMA协议在这些参与节点之间进行通信。这可以实现非常低的延迟数据传输和低的用于进程间通信(IPC)应用程序的CPU的使用率。远程数据放置是直接在源节点和目标节点之间实现的，因此避免了数据复制，并最大限度地减少了操作系统的参与。这些因素共同减少了总体延迟和CPU开销(图1.2)。

任何IB设备都可以与一台或多台IB设备或交换机相连。这种连接可以使用一个或多个链路。

### 网络拓扑

#### 1.3.2.1通道适配器

IBA定义了驻留在服务器或I/O系统中的两种类型的适配器。如果适配器驻留在主机系统(例如，服务器)中，它被称为主机通道适配器(HCA)。如果适配器位于存储目标系统中，则称为目标通道适配器。

通道适配器为操作系统和应用程序提供到物理端口的连接。HCA为操作系统提供接口，并提供IB定义的所有谓词接口。谓词是应用程序和通道适配器提供的功能之间的抽象接口。每个适配器可以有一个或多个端口。每个端口通过“虚拟通道”(VL)进一步区分流量。每个VL可以独立控制流量。如图1.3所示，DMA可以从本地和远程应用程序发起。

通道适配器携带一个称为全局唯一标识符(GUID)的唯一地址。适配器的每个端口还分配了一个唯一标识符，即端口GUID。GUID由适配器供应商分配。给定子网的管理实体称为子网管理器(SM)，它为Channel Adapter的每个端口分配本地标识符(LIDs)。

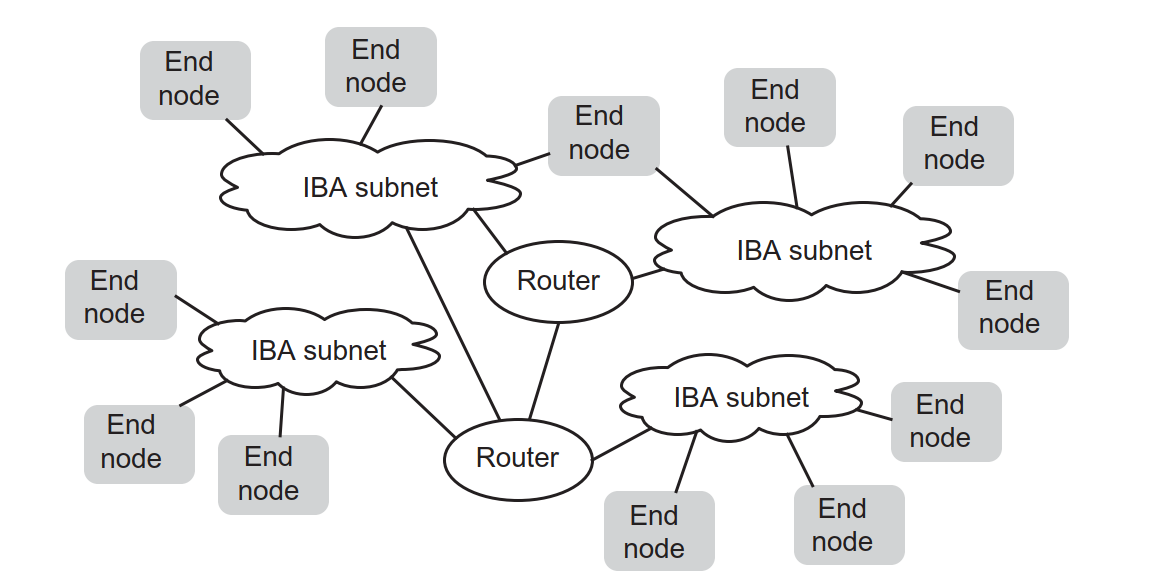


图 1.2 IB网络组件

图示

描述已自动生成

图 1.3 IB网络适配器

图示

描述已自动生成

图 1.4 IB交换机

#### 1.3.2.2交换机

一台交换机包含多个IB接口。它根据每个数据包的本地路由头(LRH)中的目的地址在适配器端口之间转发数据包。交换机根据转发数据库中的配置，将单播数据包转发到单个目的端口，将多播数据包转发到多个端口。交换机使用的转发数据库由SM配置。交换机只是转发数据包——它们不修改数据包的报头(图1.4)。

#### 1.3.2.3路由器

与交换机类似，路由器将数据包从源转发到目的。但是，路由器根据全局路由头GRH (global routing header)转发报文。它们在从一个子网转发到另一个子网时修改LRH。每个子网用子网前缀标识。路由器使用IPv6指定的协议交换路由信息(图1.5)。

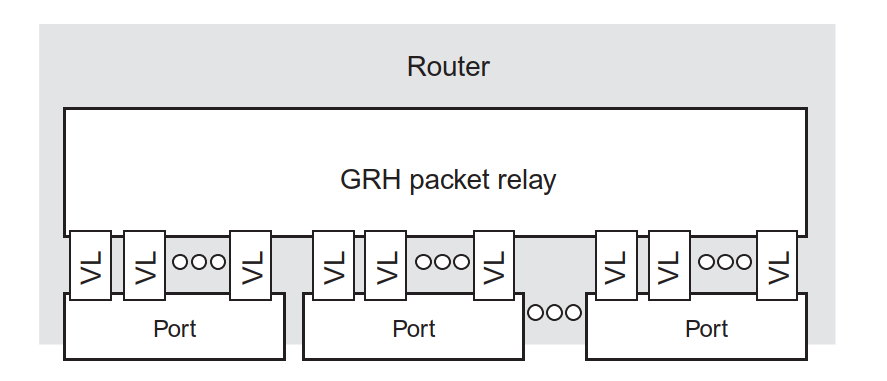


图 1.5 IB路由器

源节点指定路由器的LID和报文要发送到的目的地的GID (global identifier)。每个路由器使用子网信息和路由表将数据包转发到下一个路由器。路由一直进行到报文到达目的子网。最后一个路由器使用与目的GID相关联的本地ID将数据包转发到最终目的地。

#### 1.3.2.4子网管理器

SM在子网中配置本地组件。它为子网内的所有节点提供LIDs，并为子网内的交换机提供转发信息。短信通过子网管理代理(SMAs)与子网内的所有节点通信。每个IB节点都需要实现SMA。

一个子网中可以有多个SMs，但在给定时间只能有一个是活动的。所有非活动/备份短信都维护活动短信的转发信息副本，并在活动短信停机时使用该副本继续向子网提供管理服务。

## 通信机制

### 通信服务(传输)

IBA根据应用程序的需要在IB节点之间支持不同类型的通信机制。

#### 1.4.1.1可靠的连接和可靠的数据报

为了实现可靠的通信，数据通过序列号和确认消息(ACK/NAK)的组合可靠地传递。一旦检测到错误或数据包丢失，源可以通过重传数据包来恢复，而不需要用户应用程序的参与。这种模式保证消息包只传递一次。当应用程序需要依赖底层传输来保证将消息传递到目的地时，使用此模式。这种模式释放了应用程序，以防止底层媒体或传递机制的不可靠性。

RC (Reliable connection)模式通过源节点和目的节点之间的直接专用连接，在节点之间提供可靠的数据传输。

RD (Reliable datagram)模式为任何终端节点提供可靠的分组消息传递，源端和目的端节点之间不需要建立专用连接。这是一个可选模式。

#### 1.4.1.2不可靠的数据报和不可靠的连接

不可靠模式适用于对丢包不敏感或本身能够处理丢包，但希望快速传输数据的应用程序。在这种模式下，不能保证数据从源节点传输到目的端节点。

在不可靠数据报(UD)模式下，数据不需要建立任何连接就可以从源节点发送到目的端节点。不能保证包的发送。在这种模式下，不会发现数据丢失。

在不可靠连接(UC)模式下，在源端和目的端节点之间建立一个专用的连接，在没有传输保证的情况下进行消息的传递。错误(包括序列错误)会被检测并记录下来，不会被通知回源端节点。

#### 1.4.1.3原始IPv6数据报和原始以太类型数据报

这是一种特殊的UD模式，只使用本地传输头信息。这种模式被非IBA传输层用于跨IB网络的数据隧道。

### 寻址

节点间的IB通信需要对每个可寻址实体(节点、卡、端口、端口内的队列对(QP)等)进行唯一标识，以便数据包能够适当地传递。这些数据包可以是子网内通信的一部分，也可以属于通过路由器跨子网的流。流可以是单播的，其中通信只在两个可寻址实体之间进行。多播流用于多个实体之间的通信(图1.6)。

LID:是一个16位的单播或组播标识符，在子网内是唯一的;不能用于子网间路由。一个LID由一个SM分配。盖子包含在每个数据包的LRH中。

GID: GID是一个128位单播或多播地址，并且是全局唯一的，这允许它用于跨子网路由数据包。GID是具有IBA定义的附加限制的有效IPv6地址。GID分配范围从默认分配(从制造商分配的标识符计算)到SM分配的地址。

图示

描述已自动生成

图 1.6 IB寻址范围

单播标识符:单播LID或GID标识单个可寻址实体。以单播标识符为地址的数据包将被发送到单个端端口。

组播标识符:组播LID或GID标识一组可寻址的端端口。发送到多播标识符的数据包必须传递到属于该标识符的所有端端口。

### 数据包格式

在IB网络中定义了两类数据包。

IBA数据包:携带传输头的IB数据包在IBA 结构上路由，并使用本地IBA传输设施。

原始数据包:这些数据包通常用于在IB网络上传输非IBA数据包。所以这些包不包含IBA传输头。

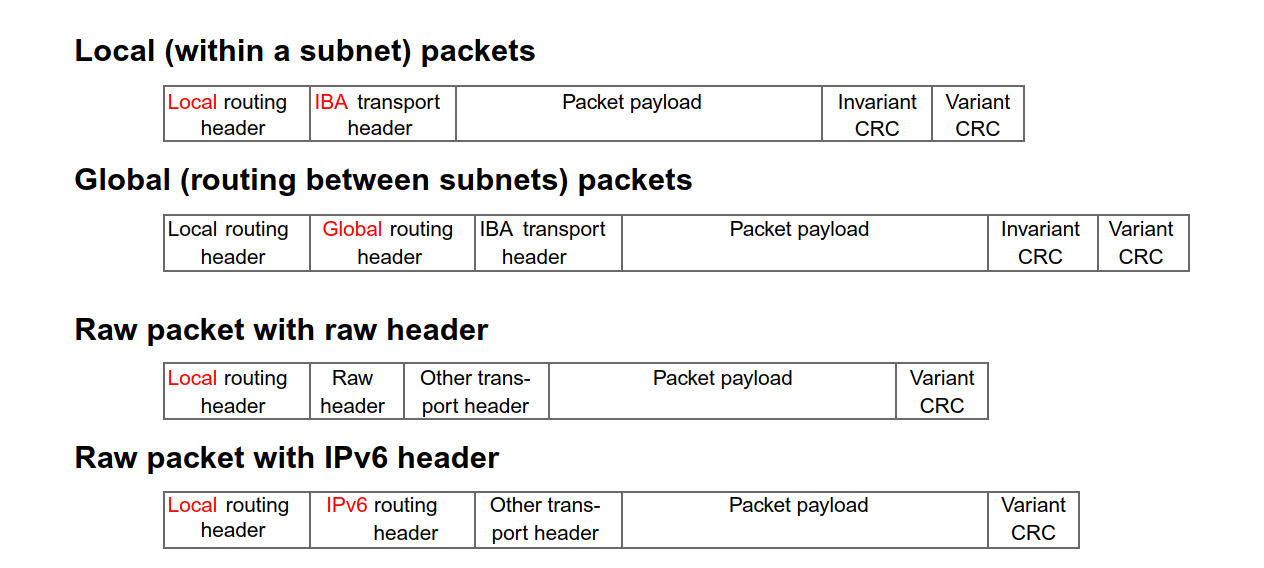


图 1.7 IB寻址范围

IBA定义的数据包格式如图11.7所示。

所有的报文都需要一个本地路由头(8字节)。该报头用于转发本地子网内的数据包[11]。

所有需要路由到不同子网的数据包和所有组播数据包都需要GRH(40字节)。LRH中的link next header字段表示报文中存在GRH头。

Raw报文只包含一个LRH和一个Raw或IPv6路由头。

IBA在报文中支持两种CRC(循环冗余校验)。不变CRC(4字节)涵盖了数据包在从源节点到目的节点的传输过程中不发生变化的所有字段。VCRC (Variant CRC)， 2字节，覆盖报文中的所有字段。每个IBA包携带不变的CRC，然后是VCRC。原始数据包只携带VCRC。

## 分层体系结构

分层体系结构允许用不同的组件构建解决方案，这些组件可以通过实现的灵活性进行互操作，同时由于层之间定义良好的接口而具有操作的正确性。它还为系统中的不同功能块提供了架构清晰度和分离，这有助于应用程序通过各种协议、网络和物理连接选项相互通信。这提供了各种可能的应用程序部署，而不需要对实现进行自上而下的更改。在改变物理介质的情况下，可以在铜缆或光纤电缆上运行系统，而不需要改变物理层之上的任何协议。类似地，可以在一个子网内或跨多个子网运行应用程序，而应用程序不会意识到通信系统之间的网络分离(图1.8)。

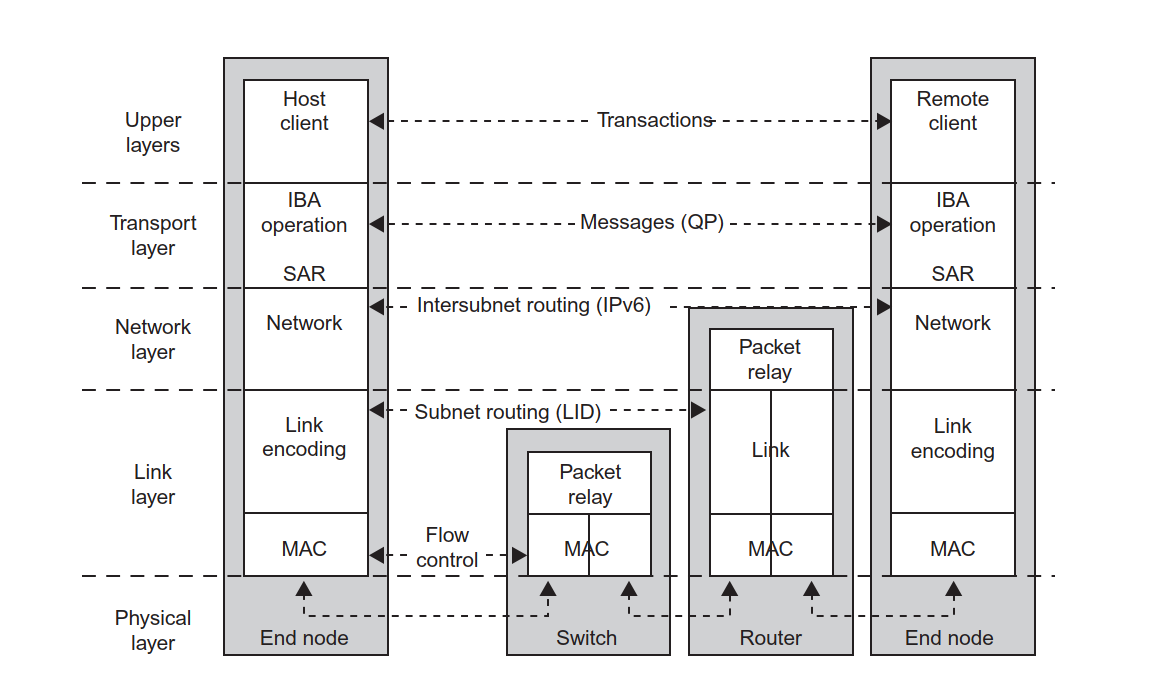


图 1.8 IB层的架构

### 物理层

物理层定义了在系统之间的物理连接上实际比特是如何流动的。系统可以通过背板或光纤或铜介质相互连接。电缆的长度可以改变，机械连接器的长度也可以改变。IB的物理层定义了这种连接的电气和机械方面。

物理层负责从链路层接收(控制和数据)字节，将它们以电或光的形式发送给链路对等体，然后将接收到的字节发送给位于接收端的链路层。它向链路层提供了关于连接到链路对等体的实际物理介质的透明性。

物理层还负责底层物理介质与链路对等体之间的速度和宽度协商。

IBA定义了四种类型的物理连接选项来连接IB设备。它提供了所有这些的电气、光学和机械规格。以下列出了物理连接选项:

1. 底板的港口

2. 铜的港口

3.光纤端口

4. 有源电缆端口

#### 1.5.1.1数据包格式和链路宽度

数据包由电线上的特殊符号分隔，称为SDL(数据包的起始分隔符)和EGP(好数据包的结束分隔符)或EBP(坏数据包的结束分隔符)。链路是由两个节点之间的多个“通道”连接形成的。单通道端口称为1X端口。类似地，4通道或12通道的端口分别称为4X或12X。对于具有多个通道的端口，数据包在所有通道上都是“字节条纹”的(图1.9)。

#### 1.5.1.2速度和宽度协商

IBA运行链路的速度不同，随着新一代链路的发展，运行链路的速度更高。在SDR(单数据速率)中，信令速率为2.5 Gbps。在接下来的几代中，速度已经增加到5Gbps (DDR)，10Gbps (QDR)等。图1.10显示了IB链路层的速度序列(也预测了未来的速度)[12]。

### 链路层

IB体系结构中的链路层定义了跨物理连接发送和接收数据包的机制。这些机制包括寻址、缓冲、流控制、QoS、错误检测和切换。第1.4.2节详细讨论了寻址。

#### 1.5.2.1报文转发

链路层定义IBA子网内报文的转发。在子网内，报文通过LIDs进行转发。LRH的格式如图1.11所示。这些标识符由SM在设备上配置，如1.3.2.4节所述。交换机使用目标LID查找需要转发给定数据包的目标端口。IBA要求在流(子网内相同源LID和目的LID之间的数据包)中的单播数据包中保持有序的数据包传递。

图示

描述已自动生成

图 1.9 跨通道的字节延展数据

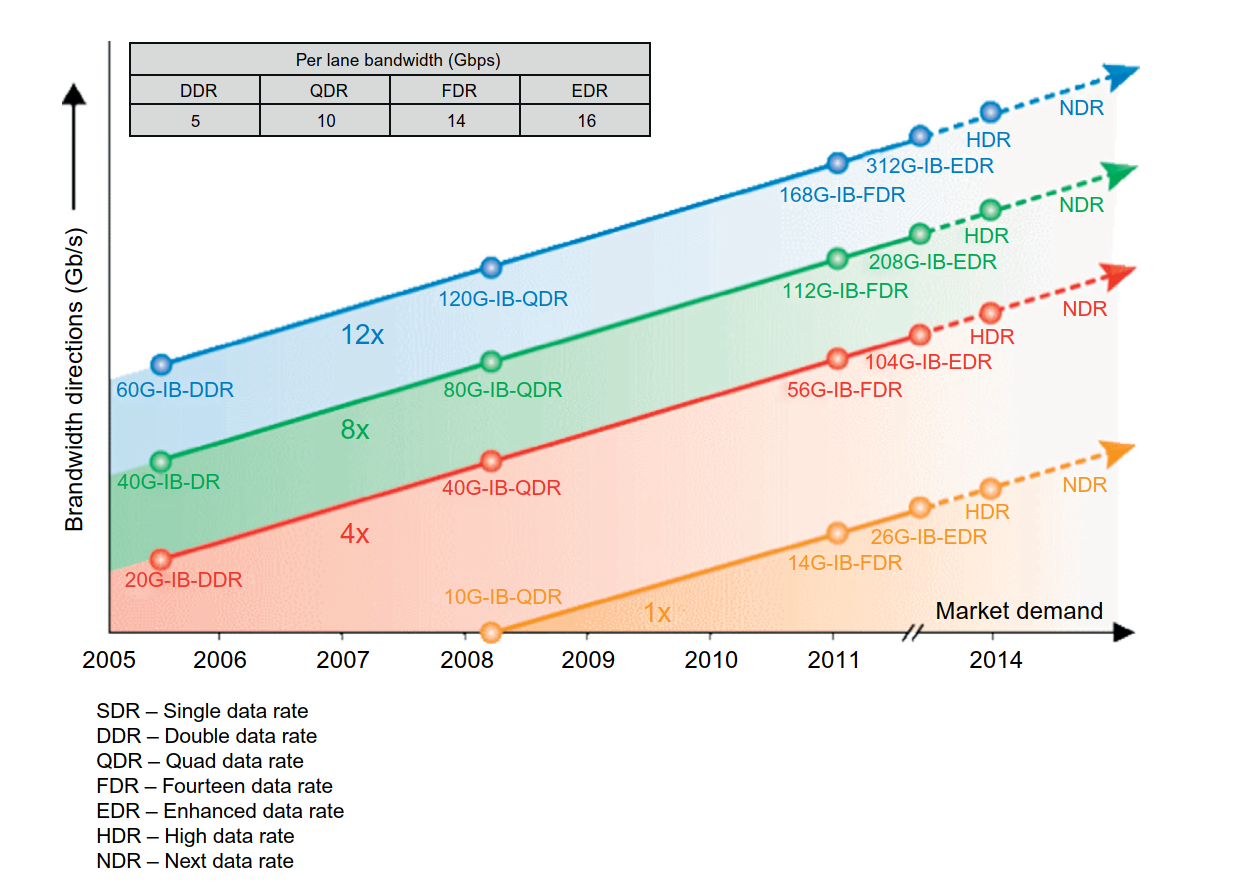


图 1.10 IBTA信令速率路线图

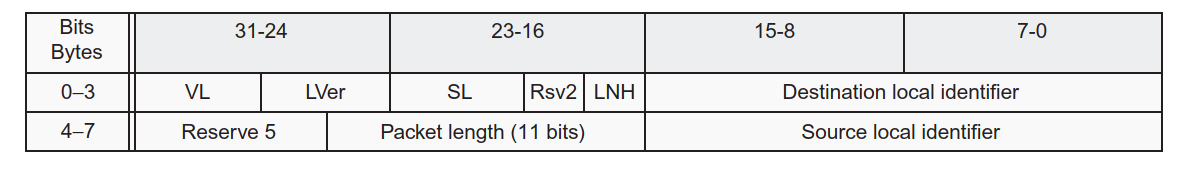


图 1.11 本地路由头(LRH)

这同样适用于组播和广播报文。然而，在节点接收到的单播和多播数据包之间，顺序要求并不适用。

#### 1.5.2.2数据完整性

IBA使用CRC字段保证数据流经网络时的完整性。32位不变CRC包含包中所有端到端不变的字段。该字段不需要在每一跳重新计算。该字段不存在于原始数据包中。在网络中变化的字段由一个16位的VCRC覆盖。VCRC在每一跳处计算，以捕获任何可能在路由过程中导致报头修改的数据完整性折衷。该字段存在于所有报文中，包括原始报文。

#### 1.5.2.3虚拟通道

IB允许通过称为VLs的抽象在电线的物理连接上创建虚拟链路。就链路级流控制等机制而言，每个VL都可以相互独立地运行。这允许在给定的物理端口或物理链路上区分不同的“流”。流可以分组到属于特定的“VL”，一致的流量工程原则可以跨多个设备应用，为给定的VL提供一定的QoS。

每个包在链路头的4位VL字段中携带关于它属于哪个VL的信息。每个VL都被提供了独立的缓冲资源，以保证它们不会相互干扰。

每个VL的链路级流量控制确保一个VL中的流量不会受到另一个VL上断言的流量控制的影响。IBA在每个链路上定义了15个VL。VL15被认为比VLO具有更高的优先级。与低优先级流相比，高优先级流可以更快地得到服务。带宽/资源可以根据流的VL规则分配给流。管理流量使用优先级最高的VL,VL15。要求每台设备支持VLO和VL15。

#### 1.5.2.4服务层级

除了VL之外，IBA还定义了一种为流服务级别(SL)分配QoS标识符的机制。这个4位字段包含在LRH中，它标识子网内给定流的SL。数据包中的这个字段在数据包通过网络时不会被修改。SL的实际含义留给实现——然而，它的目的是由产品使用，为SM规定的流提供流量差异。

IBA定义了将SL字段映射到给定端口的VL的机制。这种SL到VL的映射是通过映射表实现的，它允许IB节点根据映射表定义的规则和VL分配的固有期望为流提供QoS。例如，映射到高优先级VL的SL在给定端口上会获得更高的优先级。

由于每个SL在给定端口上获得不同的调度，因此不同SL之间不维护排序。

#### 1.5.2.4缓冲和流量控制

IBA提供了一种机制，通过基于缓冲区到缓冲区信用的流量控制机制保证数据包在链路上的无损传输。这要求每个接收方向发送方提供有关缓冲区可用性的信息，以便发送方能够通过线路将数据包传递给接收方。由于有一个保留缓冲区等待到达的数据包，因此不存在接收方由于拥塞而丢弃数据包的情况。每个VL都需要在给定端口上有单独的缓冲。这允许在每个VL上使用单独的流量控制。

IBA为接收方定义了一种机制，用于通知发送方在给定时间点允许传输的数据量。IBA还指定了协议，以确保用于交换关于流控制本身的信息的通信协议是无错误的，并且可以从错误条件(如果出现错误条件)中恢复。发射器和接收器定期重新同步它们的信息，以纠正给定VL上关于信用可用性的任何不一致的信息。

除了链路级流控制之外，IBA还指定了拥塞控制机制，允许网络中拥塞的端口请求流的实际源减慢速度(与在链路级流控制中仅传输前一个节点的端口相比)。IBA指定了一种机制，在这种机制中，VL上的拥塞可以被检测到，拥塞的转发通知(FECN:转发显式拥塞通知)由一个针对违规数据包的开关标记。这个位被目的地解释，并作为一个特殊的管理数据包转过来，用于向错误流的源发送向后通知(BECN:向后显式拥塞通知)。然后源解释这个数据包，并暂时降低数据注入到给定拥塞目的地的速率(随着时间的推移恢复原始的注入速率)。

### 网络层

网络层提供跨多个子网的路由。它指定跨IBA子网转发单播和多播数据包。这种路由可以通过符合IBA和非IBA(例如，IP)规范的路由器来完成。图1.12中GRH中提供的字段可以用于这种路由。通常，这些字段包括SGID、DGID、TClass和流标签(这些字段可以很容易地映射到IPv6词汇表中，这是有意为之)。SGID (Source Global Identifier)和DGID (Destination Global Identifier)是128位字段，可以映射到IPv6地址。路由的工作原理与IP路由非常相似，其中源子网中的目的本地标识符(DLID)将映射到本地路由器地址，目的路由器将通过将数据包的DLID更改为该子网中的最终目的端口来确保数据包被传递到目的节点。

表格

描述已自动生成

图 1.12 全局路由头(GRH)

### 网络层

传输层为上层协议(ULPs)(和应用程序)提供了一个接口，通过网络层使用QP进行发送和接收操作，在子网内和子网间进行通信。它负责使用应用程序所需的传递特征(例如，可靠与不可靠以及连接与数据报)将数据有效负载从源端节点传递到目标端节点。传输层根据传输报头中的信息向正确的QP发送数据包。传输层还负责向ULP提供分段和重组服务。它根据底层网络层支持的最大传输单元，将传输路径中的消费者数据分割为大小合适的有效负载。在传输过程中，每个段都用报头和CRC封装。接收后，QP在内存中重新组装指定ULP缓冲区中的所有段。数据的实际传输及其交付依赖于给定QP所配置的服务类型。关于这些机制的细节将在第1.5.1节中讨论。