



# 盛科 CTC8180 分布式路由方案

版本 R1.0  
日期 2020-12-14

版权所有 © 盛科网络（苏州）有限公司。保留一切权利。

未经盛科网络（苏州）有限公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式和任何方法传播。



盛科商标，服务标志和其他盛科标志均为盛科网络（苏州）有限公司拥有商标。盛科交换机系列产品和芯片系列产品的标志均为盛科网络（苏州）有限公司商标或注册商标。未经盛科书面授权，不允许使用这些标志。

本文档提及的其他所有商标和商业名称，由各自的所有人拥有。

## 注意

您购买的产品、服务或特性等应受盛科网络商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，本公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

## 盛科网络（苏州）有限公司

地址 江苏省苏州市工业园区星汉街 5 号（腾飞新苏工业坊）B 幢 4 楼 13/16 单元

电话 86-512-62885358

传真 86-512-62885870

网址 <http://www.centecnetworks.com>

邮箱 [support@centecnetworks.com](mailto:support@centecnetworks.com)

## 内容目录

<b>1 分布式路由原理概述 .....</b>	<b>7</b>
1.1 分布式路由简介 .....	7
1.2 芯片原理.....	7
1.2.1 芯片堆叠原理 .....	7
1.2.2 CFlexHeaderBasic.....	8
1.2.3 CFHeaderExtEgrEdit .....	9
1.2.4 CFHeaderExtCid.....	9
1.2.5 CFHeaderExtLearning .....	10
1.2.6 CFHeaderExtOam .....	10
1.3 CFlex Port 查找模式 .....	10
<b>2 分布式路由方案.....</b>	<b>13</b>
2.1 分布式路由整体方案 .....	13
2.1.1 整体流程 .....	13
2.1.2 三层接口恢复 .....	13
2.1.3 最长匹配和 PBR.....	16
2.1.4 Overlay/Underlay .....	16
2.1.5 Misc.....	17

## 表格目录

表 2-1 Source Interface 的属性表 .....	14
-----------------------------------	----

## 图形目录

图 1-1 CFlexHeader .....	8
图 1-2 CFlex Port 查找模式 .....	11
图 2-1 LPM 路由查找流程.....	13

## 修订记录

日期	版本号	说明
2020-12-14	R1.0	初始发布

# 1 分布式路由原理概述

## 1.1 分布式路由简介

在分布式交换机的应用场景中，通常会有处理业务相关的线卡，和处理不同业务线卡之间交换的交换网板。在常规的应用中，业务相关的转发表项都是在线卡上，在交换网板上只有板间转发的信息，没有业务转发表。而分布式路由则是考虑把路由进行分拆，主机路由在线卡处理，非主机路由在交换网板上处理。

## 1.2 芯片原理

分布路由的技术的实现，需要依赖于堆叠的 CFlex 来实现芯片的互联。芯片链之间通过 CFlex Port 连接，芯片之间通过 CFlex Header 通信。

通过在 CFlex 技术的基础上，扩展了在 CFlex Port 支持转发表的查找，这样就可以使用分布式路由技术来扩展路由表的规格。

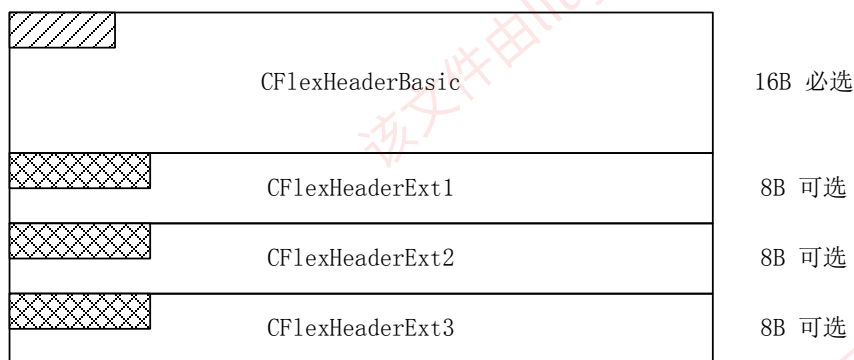
### 1.2.1 芯片堆叠原理


CTC8180 在保留原有 CFlexHeader 结构基础上，新增一种全新的 CFlexHeader 结构。新增的 CFlexHeader 结构有两部分组成。

一部分是固定的 16 字节 CFlexHeaderBasic，这部分是必选 CFlexHeader，通常应用只要这个头部就足够了。

另外一部分是有多种类型扩展头部组成，每种类型的扩展头部支持一种或者一部分业务特性。每个扩展头长度为 8 字节。扩展头部不是必选，可以根据业务按需携带到 CFlexHeaderBasic 后面。目前支持四种类型的扩展头部，分别是：

- CFlexHeaderExtEgrEdit: egressEdit 所需信息
- CFlexHeaderExtLearning: macSa 学习所需信息
- CFlexHeaderExtOam: oam 所需信息
- CFlexHeaderExtCid: 其他业务所需信息



 extHeaderLen[2:0], 扩展头长度, 以8字节为单位


 extHeaderType[3:0], 扩展头类型

图1-1 CFlexHeader

上图为 CFlexHeader 结构示意图。一个完整的 CFlexHeader。为了减小开销, CFHeaderExtLearning 和 CFHeaderExtOam 两种扩展头部不会同时携带, 一旦需要携带 CFHeaderExtOam, 则自动忽略 CFHeaderExtLearning。

## 1.2.2 CFlexHeaderBasic

CFlexHeaderBasic 总共 16 字节, 为必选字段。主要包含下列字段:

- extHeaderLen[2:0], 扩展头长度(不含必选头部长度), 以 8 字节为单位。比如后续有 2 个扩展头, extHeaderLen 为 2
- svlanTpidIndex[1:0], SVLAN 的 TPID 索引值
- destMap[21:0], 表示目的地信息。如果目的地是组播, 包括{isMcast, 5'b0, destId[15:0]}, isMcast 置 1, destId[15:0] 表示组播组 Id; 如果目的地是单播, {isMcast, 2'b0, isToCpu, destChid[6:0], destId[8:0]}, isMcast 置 0, isToCpu 表示该报文是否是送到 CPU 的, destChid[6:0] 表示目的芯片号, destId[8:0] 表示单播目的端口号
- macKnown, 表示在 Ingress 芯片二层查找时是否匹配到 VLAN 默认转发条目, 1 表示匹配了具体的二层转发表项, 0 表示匹配到了 VLAN 默认转发条目。对于非二层转发的报文, macKnown 默认置 1
- outerVlanIsCVlan, 用于指示 egress chip 如何解封装携带两层 Tag 的报文。如果 outerVlanIsCVlan 置 1, 则外层 Tag 会被解析成 C-Tag
- srcVlanPtr[12:0]
- sourcePort[15:0], 报文进入系统的原始的源端口号 globalSrcPort
- priority[3:0], 报文转发优先级。Priority 从之前的 6 个 bit 减少为 4 个 bit.
- color[1:0], 报文颜色
- packetType[2:0], 报文类型, 比如是普通以太网报文
- headerHash[7:0], Hash 值用于端口聚合选择成员端口
- logicSrcPort[15:0], 逻辑源端口号



- sourcePortIsolateId[6:0], 源端口的端口隔离组号, 从之前的 6bits 增加为 7bits
- macLearningEn, 表示是否需要做 MACSA 的查找
- fromLag, 表示报文来自 LAG 口
- fid[13:0], 用于 MAC 地址学习

### 1.2.3 CFHeaderExtEgrEdit

CFHeaderExtEgrEdit 是可选扩展头部, 总共 8 字节, 如果报文做 egress edit, 需要携带这个头部。主要包括下列字段:

- extHeaderType[3:0], 扩展头部类型, CFHeaderExtEgrEdit 扩展头类型为 1
- nextHopPtr[17:0], 用于读取 egress chip 上 DsNextHop 表
- ttl[7:0], 原始报文的 TTL 值
- srcDscp[5:0], 进来报文的 DSCP 值
- egressEditEn, 如果置 1, 表示 egress edit, 报文编辑将在出接口所在芯片上进行
- ecmpHash, 用于新头部编辑的时候, 映射 VxLAN 的 UDP source port, 或者 NvGRE 的 GRE key

报文使用 Ingress 编辑或者 Egress 编辑, 取决于具体转发业务, 基本二三层转发只需要使用 Ingress 编辑, 不需要携带 CFHeaderExtEgrEdit, 减小开销。可以通过转发表中的 BypassIngressEdit(eg: DsFwd.bypassIngressEdit)字段置 1, 使能 Ingress 编辑; 如果 BypassIngressEdit 置 0, 表示使用 Egress 编辑。

### 1.2.4 CFHeaderExtCid

CFHeaderExtCid 是可选扩展头部, 总共 8 字节, 携带 categoryId, stacking 拓扑发现, 报文截断等相关的属性。主要包括下列字段:

- extHeaderType[3:0], 扩展头部类型, CFHeaderExtEgrEdit 扩展头类型为 2
- neighborDiscovery, 用于 stacking 拓扑发现。通常只有 CPU 下发的报文才可能会被置上这个 bit, 普通数据业务跨芯片转发的时候不会置这个 bit
- isLeaf, 用于水平分割检查
- isSpanPkt, 表示报文是否是被镜像出来的
- truncateLenProfId, DsTruncationProfile 表的索引, 用于对报文做截断
- i2eSrcCidValid, 置 1 表示源 categoryId 字段有效
- i2eSrcCid[7:0], 源 categoryId
- pktWithCidHeader, 置 1 表示原始报文中携带这 categoryId Tag
- terminateCidHdr, 置 1 表示出方向无需携带 categoryId Tag
- bypassCFlexSrcCheck, 置 1 表示在 Stacking 环上转发的时候, 该报文需要跳过防环检测。通常只有 CPU 下发的报文才可能会被置上这个 bit

## 1.2.5 CFHeaderExtLearning

CFHeaderExtLearning 是可选扩展头部， 总共 8 字节， 携带 MAC learning 相关的属性。主要包括下列字段：

- extHeaderType[3:0], 扩展头部类型， CFHeaderExtEgrEdit 扩展头类型为 3
- macAddr[47:0], 用于 MAC 地址学习

通过设置 EpeHeaderEditCtl.cfHeaderLearningEn 为 1， 使能携带 CFHeaderExtLearning 的功能。使能后， 对于需要做 learning 的报文， 发往 Stacking/Fabric 口的报文会将 MacSa[47:0]封装到 CFHeaderExtLearning 扩展头中。

## 1.2.6 CFHeaderExtOam

CFHeaderExtOam 是可选扩展头部， 总共 8 字节， 这里面包括一些 OAM 相关的属性。具体字段包括：

- extHeaderType[3:0], 扩展头部类型， CFHeaderExtEgrEdit 扩展头类型为 4
- dmEn, 使能 DM
- mipEn, 使能 MIP
- localPhyPort[8:0], 入端口号
- mepIndex[13:0], 用于读取 MEP 属性的索引
- oamPacketOffset[7:0], 报文头部偏移量， 用于解析 OAM 报文

## 1.3 CFlex Port 查找模式

CTC8180 支持在 CFlex Port 进行 LPM 查找和转发， 下面详细说明了实现的原理。

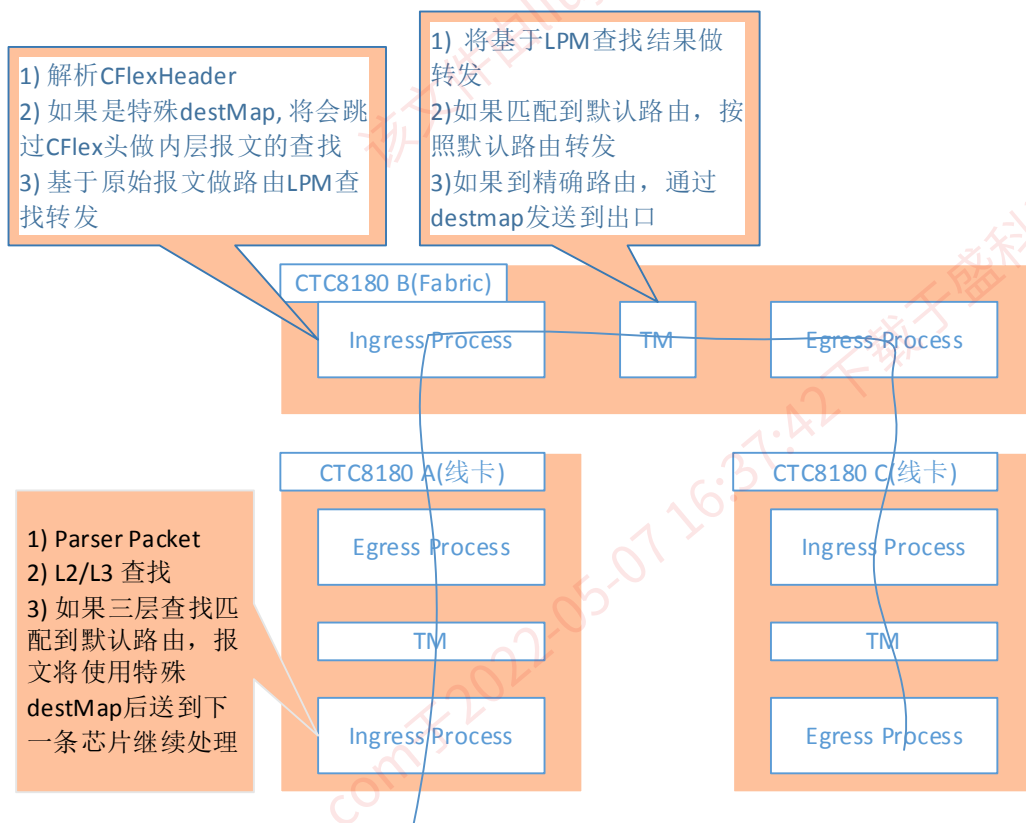


图1-2 CFlex Port 查找模式

报文从业务端口进入芯片 A 后, 首先进行报文解析, 根据配置做二三层的查表进行转发。路由报文如果匹配到精确路由, 则按照正常主机路由的下一跳对报文进行编辑和转发。如果匹配到默认路由, 得到软件配置的特殊 DESTMAP, 将报文通过 CFlex Port 携带 CFlex header 转发到芯片 B 继续做三层路由查找。

在芯片 B 上需要做些配置, 识别需要做 LPM 查找的报文, 并且能够做三层路由查找和转发:

- 识别需要做 LPM 查找: 通过配置 `IpeHeaderAdjustCtl.cFlexLookupDestMap(21,0)` 和 `IpeHeaderAdjustCtl.cFlexLookupDestMapMask` 来识别特殊 DESTMAP, 识别出后, 将跳过 CFlexHeader 对报文进行解析, 路由查找和转发。
- 获取报文转发时需要用到的 `localPhyPort`: 有两种方式得到, 当 `IpeHeaderAdjustCtl.cFlexLocalPortMode` 配置为 0 的时候, `localPhyPort` 由 `IpeHeaderAdjustPhyPortMap.localPhyPort(6,0)` 配置, 即为 stacking 口的端口号; 当 `IpeHeaderAdjustCtl.cFlexLocalPortMode` 配置为 1 的时候, `localPhyPort = DESTMAP(7,0)`。
- 在去掉 CFlex header 后在 LocalPhyPort 上的行为和普通端口进来的 PIPE line 流程相同。
- 三层接口的获取方式有下面三种方式来实现:

- ◆ 通过 localPhyPort 配置三层口，使能三层查找: localPhyPort 对应的 DsSrcPort.routedPort 置 1. DsSrcPort.interfaceId 设置为对应的 L3ifId. 配置 L3ifId 对应的 DsSrcInterface 表的属性。
- ◆ 通过 CFlex header 中的 SrcVlanPtr 来恢复 L3ifId。寄存器 lpeHeaderAdjustCtl.cflexUserHeaderVlanPtr 控制把 SrcVlanPtr 来给 uservlanPtr，从而在后续处理逻辑中得到 L3if。
- ◆ 通过 LogicSrcPort 查找 scl 得到 L3if。寄存器 lpeHeaderAdjustCtl.cflexUserHeaderLogicSrcPort 控制从 CFlex header 中得到入芯片的 logicSrcPort，在后续 SCL 部分可以根据 logicSrcPort 查找 SCL 得到 L3if。

报文匹配到路由条目后，根据对应的下一跳的配置对报文进行编辑，然后将 destChipId 和 destPortId 封装到 CFlexHeader 的 destMap 中，将报文送到最终目的芯片 A 上。目的芯片 A 根据 CFlexHeader 中携带的信息将报文从最终的业务端口转发出去。

# 2 分布式路由方案

## 2.1 分布式路由整体方案

### 2.1.1 整体流程

基于上文的 CFlex 技术，本文以三片 CTC8180 为例实现分布式路由，实现路由表项的在 Fabric 芯片上的扩展。

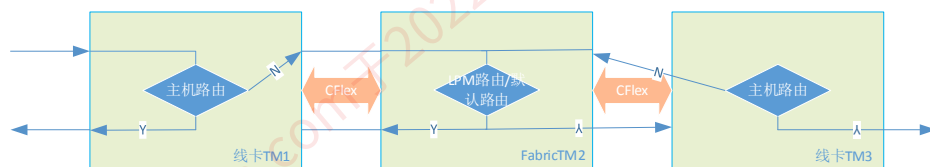


图2-1 LPM 路由查找流程

我们定义线卡 1 芯片为 TM1，Fabric 芯片为 TM2，线卡 2 芯片为 TM3。

在每个芯片中的路由转发行为定义如下：

TM1：查找路由报文，匹配到 host 路由的话，根据 destmap 转发出去；匹配到默认路由，通过 CFlex Port 送给 TM2，并且在 CFlex header 中通过 destmap 携带 TM2 相应的入口 Port，该 Port 需要为 L3 interface 属性集合的代表，典型的表示 VRF 信息。

TM2：通过 LocalPhyPort 或者报文 Vlan 恢复 L3 Interface，在对应的 Vrf 内查找路由，如果查找到 LPM 路由，根据 destmap 转发到 TM1 或者 TM3。否则匹配到默认路由，根据默认路由的配置转发。

TM3：TM3 上的转发行为和 TM1 相同。

### 2.1.2 三层接口恢复

报文在 Fabric 芯片的 CFlex Port 上接收到做查找的时候，目前有两种方式恢复原有的 L3 interface。

对于非 tunnel 的报文或者 tunnel encap 的方向，在 CFlex header 中携带 srcVlanPtr 来标识入芯片对应的 L3Ifid，在 Fabric 上根据 srcVlanPtr 得到有效的 L3Ifid。

对于 tunnel decap 的情况下，在入芯片上 decap 的时候，需要出 logicSrcPort, logicSrcPort 和 tunnel Vrf 是对应的，并且会通过 Cflex header 携带到 Fabric 上，Fabric 上通过 SCL 查找 logicSrcPort 的 key 表来获取 Vrf 的信息、

表2-1 Source Interface 的属性表

Field Name	Description
categoryId[7:0]	Category ID of this interface.
categoryIdValid	If set, categoryId is enabled on this interface.
contextLabelExist	If set, it indicates that context label is exist on this L3 interface. It will do process about context label when MPLS type is UpStream.
dscpPhbPtr[3:0]	PHB base index for DSCP. {dscpPhbPtr[3:0], Dscp[5:0]} will be used as the index of lpePhbDscpMap.
exception3En[15:0]	If not set, the corresponding exception[3] sub exception will be reset
igmpSnoopEn	If set, IGMP/MLD is enabled on this layer3 interface.
interfaceLabelValid	If set, per-interface label space. Otherwise, per-platform label space.
ipPublicLookupEn	If set, ip routing lookup will be performed in public routing table.
ipmcUseVlan	If set, the IP multicast packet will use S-TAG VLAN ID for lookup, instead of VRF ID.
I3IfStatsPtr[15:0]	It is index to I3 interface statistic.
I3IfType	If set, the IP interface is an internal interface, else it is an external interface. (Used for NAT)
mplsEn	If set, enable MPLS label switch on this interface.
mplsLabelSpace[11:0]	It is MPLS label space for the interface.
mplsSectionLmEn	If set, it indicates that MPLS section LM is enabled on this interface.
phbUseOuterInfo	If set, PHB operation is based on the tunnel header in

Field Name	Description
	case of tunnelling process.
profileId[6:0]	Index of DsSrcInterfaceProfile.
routeAllPackets	If set, routing operation is enforced on all IP unicast packets; otherwise, only IP unicast packets with router MAC will be routed and other IP unicast packets will be discarded.
routeDisable	If set, disable routing operation on this interface.
routeLookupMode	If set, VRF ID will be used for IP lookup.
routerMacProfile[5:0]	Router MAC index used for router MAC lookup.
routerMacSelBitmap[7:0]	Select the router MAC used on interface, up to 8 router MAC.
rpfPermitDefault	If set, default route is valid for loose mode RPF check.
rpfType	If set, loose RPF check is performed; otherwise strict RPF check is performed.
trustDscp	If set, priority and color in further processing will be mapped from DSCP of the packet.
v4McastEn	If set, enable routing operation for IPv4 multicast packets on this interface.
v4UcastEn	If set, enable routing operation for IPv4 unicast packets on this interface.
v4UcastSaType[1:0]	Type code to decide which of the following operation is performed in IPv4 SA lookup: 0x0: No operation; 0x1: RPF 0x2: NAT 0x3: PBR
v6McastEn	If set, enable routing operation for IPv6 multicast packets on this interface.
v6UcastEn	If set, enable routing operation for IPv6 unicast packets on this interface.



Field Name	Description
	interface.
v6UcastSaType[1:0]	Type code to decide which of the following operation is performed in IPv6 SA lookup: 0x0: No operation; 0x1: RPF 0x2: NAT 0x3: PBR
vrflid[12:0]	It indicates the VRF ID of virtual route forwarding.

分析 L3 Source interface 的属性，发现需要在查出路由表的时候处理的特性只有基于 Port 的 RPF check, NAT, 其他相关属性可以在线卡芯片上实现。

基于 Port 的 RPF check 和 NAT 的特性目前没法支持。

### 2.1.3 最长匹配和 PBR

路由查找的是需要遵循最长匹配的原则的。在分布式路由方案中，由于 LPM 路由分布在芯片 TM2 中，主机路由在 TM1 和 TM2 中，所以能够满足最长匹配的需求。

在路由查找的时候，存在策略路由和 LPM 路由优先级的选择问题，目前是使用 ACL 来实现策略路由功能的。

对于前策略路由可以在线卡 TM1 或者 TM3 上配置 ACL 条目。

对于后策略路由，由于我们在 TM2 的 CFlex Port 上 PIPE 流程和网络口 localPhyPort 进来相同，因此我们可以在 TM2 的 CFlex Port 上映射出的 localPhyPort 上使能 ACL 功能，对只有 hit 到 default 路由的报文进行后策略。

### 2.1.4 Overlay/Underlay

对于 Vxlan 场景下路由支持，我们可以分为上行和下行两个方向来考虑：

上行情况：

当 UNI 口进来的报文匹配 L3if 的 route mac 的情况，会进行路由查找。如果匹配到 LPM 路由，直接加封装出去；否则需要送到 Fabric TM2 中进行 VRF+IPDA 路由查找，查找到 LPM 路由直接指定 TM1 或者 TM2 上的 NextHop 编辑 Vxlan tunnel。

下行情况：

在 TM1 上进行 Vxlan Tunnel 解封装，通过 Tunnel 得出 routeMac, Vrf 和 logicsrcport, 进行路由查找，当找到 LPM 路由的情况下直接发送到 TM1 出口，否则送到 CFlex Port,



这个时候先把 Vxlan 的 tunnel 去掉，到 TM2 的时候，只有内层报文，通过 SCL 恢复出来的 Vrf + IPda 查找路由，找到 TM1 的网络口。

## 2.1.5 Misc

### 1. 已知限制

- (1) uRPF 检查不支持基于 Port 的检查