

文章编号: 1001-0920(2002)05-0571-04

MPLS 应用于 ATM 环境下状态归并环路防止机制的研究

杜 荔¹, 陈忠根², 王光兴¹

(1. 东北大学 信息科学与工程学院, 辽宁 沈阳 110004; 2. 浙江大学 电气工程学院, 浙江 杭州 310021)

摘要: 分析环路对 MPLS 网络产生的影响, 在 ATM 环境下讨论为建立无环状态归并 LSP 而设计的两种环路防止算法, 通过理论分析和实验仿真将路径向量/扩散(PD)算法与着色线程(CT)算法在扩展性、对流合并的支持能力, 以及创建或重构 LSP 所需的控制信息数等方面进行比较, 得出一些有意义的结论。

关键词: MPLS; 转发等效类; 标记交换路径; 标记分发协议; TTL

中图分类号: TP 393 **文献标识码:** A

Study of state-merging LSP's loop prevention mechanism in MPLS networks under ATM environment

DU Li¹, CHEN Zhong-gen², WANG Guang-xing¹

(1. School of Information Science and Engineering, Northeastern University, Shenyang 110004, China; 2. School of Electric Engineering, Zhejiang University, Hangzhou 310021, China)

Abstract: The impact caused by the routing loop on MPLS networks is analyzed. Under the ATM environment, two kinds of loop prevention algorithms designed for establishing loop-free state-merging LSP are discussed. The path-vector/diffusion algorithm and the colored thread algorithm are compared by both theoretical analysis and experimental simulation in expansibility, the capacity of supporting the stream merging, and the numbers of control information demanded by creating or reconstructing an LSP. Some significant conclusions are obtained.

Key words: MPLS; forwarding equivalent class; label switched path; label distribution protocol; TTL

1 引言

多协议标记交换(MPLS)以其强大的技术优势发展成为业界所推崇的下一代 Internet 宽带网络技术,但在环路问题等方面还存在一些难点。系统只有具备有效处理标记交换路径(LSP)环路的能力,才

能真正体现 MPLS 在诸多方面的技术优势。

在 MPLS 网络中,显示路由与逐跳路由机制的混用可能会产生环路。事实上,LSP 是应用标记分发协议(LDP)建立起来的^[1,2]。由于 LDP 使用从三层路由协议获得的信息,而现有路由算法大多是基于分布式计算,因此在路由变化的瞬间很可能算出

收稿日期: 2001-08-07; 修回日期: 2001-10-22

基金项目: 国家自然科学基金项目(69973011)

作者简介: 杜荔(1962—),女,辽宁沈阳人,副教授,博士生,从事 IP 交换技术的研究;王光兴(1938—),男,辽宁沈阳人,教授,博士生导师,从事宽带网络技术、网络管理的研究。

一些含有环路的转发路径。一旦形成环路,不但分组无法投递到正确的目的地,而且因长期占用路由资源会造成路由拥塞,引起其它未环路分组的延时或丢弃,严重时甚至引起网络崩溃。环路至少在以下 3 种情况下尤为不利: 1) MPLS 应用在没有内部标记结构的链路层环境; 2) 环路引起的拥塞干扰了路由协议; 3) 当 MPLS 网络支持多播业务时。

2 LSP 分类

MPLS 环路控制是针对 LSP 研究消除或避免二层环路的方法。可采用的环路控制机制有环路生存、环路检测、环路防止三级。环路防止级是环路控制中的最严格级,它通过在 MPLS 域的所有标记交换路由器(LSR)间执行统一制定的协议,实现从根本上避免 LSP 环路的形成。

从环路防止的观点看, LSP 分为无状态归并 LSP 和状态归并 LSP 两种类型。无状态归并 LSP 是指 LSP 上每个 LSR 的每个入口链路都存在对应的出口链路(多个出口链路可共用同一个出口标记),如使用扩展标记分发协议(CR-LDP)或资源预留协议(RSVP)建立的显式路由 LSP。状态归并 LSP 是指 LSP 上的某个或多个 LSR 的多个入口链路可归并到单一的出口链路上,其结果是将多个入口标记合并成单一的出口标记,如使用 LDP 建立的逐跳路由 LSP^[3]。状态归并 LSP 存在一个特殊情况,即在 LSP 上的不同结点有可能同时发生多个状态归并事件,这使状态归并 LSP 的环路防止变得更加困难。

3 状态归并 LSP 环路防止算法

在 ATM 环境下, ATM-LSR 通过索引 VPI/VCI 实现分组在 MPLS 域内沿 LSP 的高速转发^[4,5]。此时,显然不便采用 TTL 域环路生存机制,当环路生存时间较长又不宜使用环路检测机制时,采用有助于改善网络扩展性的状态归并 LSP 环路防止机制是非常必要的。

3.1 路径向量/扩散(PD)算法

在 LSP 建立之前,由出口结点向其上游 ATM-LSR 发送携有自身标识的路径向量分组,途经的每个 ATM-LSR 都往路径向量分组中添入自身标识,并继续向其上游 ATM-LSR 转发。可能发生的情况有: 1) 路径向量分组顺利到达转发等效类(FEC)的入口结点,说明路径上不存在环路。于是入口结点沿

下游返回肯定确认(ACK),出口结点收到 ACK 后便启动 LDP 建立 LSP; 2) 沿途的某个 ATM-LSR 发现在收到的路径向量分组中含有自身标识,则认为该路径中存在环路。于是不再进行标记分发, LSP 建立过程终止; 3) 已建立 LSP 的路由发生变化,则发现路由变化的 ATM-LSR 启动去除可能环路的扩散计算,以防止环路的产生。某结点 RA/RB 收到来自另一结点 RB/RA 的询问、ACK 确认、NAK 确认时发生的行为如图 1 所示。

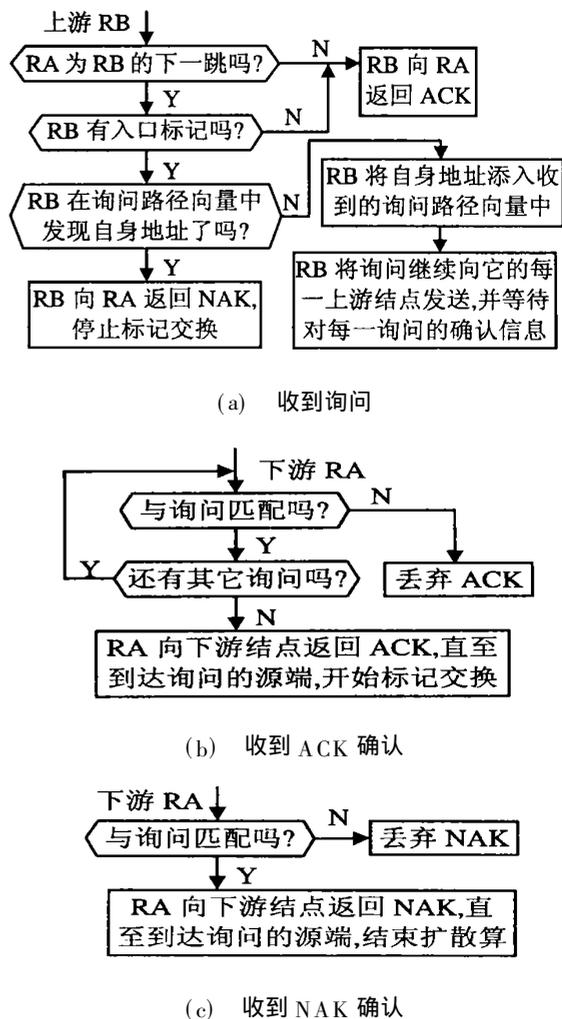


图 1 结点 RA/RB 收到来自另一结点 RB/RA 的询问、ACK 确认、NAK 确认时发生的行为

3.2 PD 算法 / 扩散计算实例

在图 2 中设: 所有 ATM-LSR(简记为 R) 中无保存路径向量的存储器; R_2, R_3, R_4 处存在归并; 所有 R 间的连接为同一 FEC 的 LSP。

当 R_3-R_4 间链路发生故障时,许多路由算法(如 RIP)很可能将 R_3 的下一跳改为 R_1 ,从而形成暂态环路。 a : 首先发现路由变化的 R_3 向其新下游 R_1 发送标记请求 Request; b : R_1 返回标记映射 Mapping;

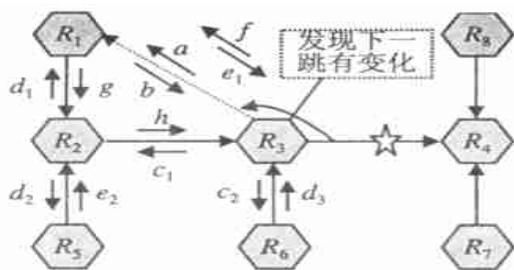


图 2 PD 算法 / 扩散计算实例

由于 R_3 有 2 个上游 R_2 和 R_6 的入口链路, 所以当它收到这个新下游 R_1 的标记映射时会启动扩散计算; c_1, c_2 : 对 2 个上游 R_2 和 R_6 分别创建一个携有相应 ATM-LSR 地址标识的询问向量 $Query(\{R_3\})$; d_1, d_2 : R_2 向 2 个上游 R_1 和 R_5 发送询问向量 $Query(\{R_3, R_2\})$; d_3 : R_6 返回对 c_2 的肯定确认 ACK; e_1 : R_1 向上游 R_3 发送询问向量 $Query(\{R_3, R_2, R_1\})$, 于是 R_3 发现路径向量中含有自身地址; e_2 : R_5 返回对 d_2 的肯定确认 ACK; f, g, h : R_3 返回对 e_1 的否定确认 NAK。当 R_3 收到 NAK 时扩散计算结束; 当 R_3 再次收到断开环路的路由改变事件时, 便又一次启动扩展计算, 自动建立起一条无环路 LSP。

3.3 着色线程(CT)算法

在 LSP 建立之前或 FEC 的某中间 ATM-LSR 发现路由的下一跳发生变化时, 由入口结点或发现变化的中间 ATM-LSR 创建一个以颜色做标识的线程, 并将线程一直向下游扩展^[6]。可能发生的情况有: 1) 出口结点收到传递来的当前颜色, 则以逐跳方式向上游返回肯定确认(ACK), 当创建着色进程的结点收到 ACK 时, 便建成一条无环路 LSP; 2) 某 ATM-LSR 发现一条以上入口链路具有同种颜色或其中一条入口链路着以由该 ATM-LSR 创建的颜色, 则判定 LSP 存在环路, 于是停止着色进程, 且不返回 ACK。此时, 除非出口链路的跳数未知, 否则, 该 ATM-LSR 将在出口链路上创建一个具有未知跳数(被认为比任何其它已知跳数都大)的新颜色线程, 并将它逐跳向下游转发。

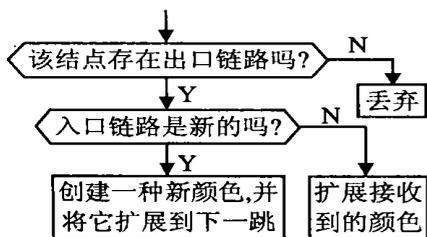


图 3 某结点在入口链路上收到一种颜色后发生的行为

某 ATM-LSR 在入口链路上收到一种颜色后发生的行为如图 3 所示。为建立无环 LSP, CT 算法还定义了扩展、反转、回收、合并和终止 5 种原始线程行为。

3.4 CT 算法实例

在图 4 中设: 所有 ATM-LSR(简记为 R) 都支持 VC 合并; 采用有序按需分配方式; 所有 LSP 都有相同的 FEC。

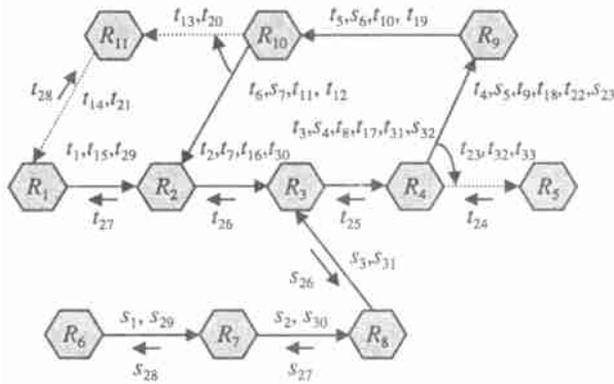


图 4 着色线程算法实例

t_1 : 入口结点 R_1 发送一个标记请求(红色, $H = 1$); s_1 : 入口结点 R_6 与 R_1 同时发送另一标记请求(蓝色, $H = 1$); t_2 : R_2 不改变颜色以(红色, $H = 2$)转发线程; s_2 : R_7 不改变颜色以(蓝色, $H = 2$)转发线程; t_3 : R_3 不改变颜色以(红色, $H = 3$)转发线程; s_3 : R_8 不改变颜色以(蓝色, $H = 3$)转发线程; t_4 : R_4 不改变颜色以(红色, $H = 4$)转发线程; s_4 : R_3 创建新颜色以(棕色, $H = 4$)扩展线程; t_5 : R_9 不改变颜色以(红色, $H = 5$)转发线程; s_5 : R_4 不改变颜色以(棕色, $H = 5$)转发线程; t_6 : R_{10} 不改变颜色以(红色, $H = 6$)转发线程; s_6 : R_9 不改变颜色以(棕色, $H = 6$)转发线程; t_7 : R_2 收到来自 R_{10} 的标记请求, 发现入口链路着以自身创建的红色, 判定红色线程形成环路, 于是终止红色线程, 创建一个新颜色且跳数未知的线程, 通过向 R_3 发送(紫色, $H = U$)请求将线程扩展; s_7 : R_{10} 不改变颜色以(棕色, $H = 7$)转发线程, R_2 将棕色线程合并到紫色线程中, 不再向 R_3 发送请求; t_8-t_{11} : R_3, R_4, R_9, R_{10} 不改变颜色分别以(紫色, $H = U$)转发线程, R_2 因发现线程环路而终止紫色线程, 又因收到线程的跳数未知, 所以不再创建新线程。

1) 设 R_{10} 将其下一跳变为 R_{11} , 则开始一个新的着色进程, 执行一次路由重构。

t_{12} : R_{10} 向 R_2 发送断路信息, 以回收紫色线程; t_{13} : R_{10} 创建新线程, 向新下游 R_{11} 发送标记请求(绿

色, $H = U$); s_{13} : R_2 收到断路信息后, 移掉终止了的入口链路 $R_{10}-R_2$; t_{14} : R_{11} 不改变颜色以(绿色, $H = U$) 向下游转发; t_{15} : R_1 改变颜色, 创建新线程(橙色, $H = U$) 向下游扩展; $t_{16-t_{21}}$: $R_2, R_3, R_4, R_9, R_{10}, R_{11}$ 不改变颜色分别以(橙色, $H = U$) 转发线程, 当 R_1 最终收到橙色线程时, 因检测到线程环路而终止橙色线程, 又因收到线程的跳数未知, 所以不再进一步创建线程。

2) 设 R_4 将其下一跳改为 R_5 , 开始新的着色进程, 执行路由重构, 自动建立起一个无环 LSP。

t_{22} : R_4 向 R_9 发送断路信息, 以回收橙色线程;
 t_{23} : R_4 创建新线程, 向新下游结点 R_5 发送标记请求(黄色, $H = U$); s_{23} : R_9 收到断路信息后, 移掉终止了的入口链路 R_4-R_9 ; t_{24} : 出口结点 R_5 发生线程反转, 向 R_4 返回标记映射; $t_{25-t_{27}}$: 标记映射逐跳地经 $R_3-R_2-R_1$ 向上游返回; $s_{26-s_{28}}$: 标记映射逐跳地经 $R_8-R_7-R_6$ 向上游返回; t_{28} : 若 R_1 在收到来自 R_{11} 的橙色线程回收信息之前收到标记映射, 则向 R_{11} 返回标记映射, 在 R_1 收到橙色线程回收信息后, 便创建一个线程以(透明色, $H = 1$) 扩展; 若 R_1 在收到橙色线程回收信息后才收到标记映射, 则移掉终止了的入口橙色链路 R_1-R_{11} ; $t_{29-t_{32}}$: R_1 收到标记映射信息便创建一个线程, 以(透明色, $H = 1$) 开始向下游扩展; $s_{29-s_{32}}$: R_6 以(透明色, $H = 1$) 开始向下游扩展线程; t_{33} : R_4 以(透明色, $H = 5$) 向下游转发线程到出口结点 R_5 。至此, LSP 创建完成。

4 两种状态归并环路防止算法在 ATM 环境下的比较

PD 算法和 CT 算法的共同之处在于: 本质上都是根据 LSP 上的结点地址及标识符做出是否发生环路的判断, 且二者都是建立 LSP 显示路由, 并可支持流合并功能。二者的不同之处主要表现在: 1) PD 算法是先以上游方式发出询问, 后以下游方式传送应答信息; 而 CT 算法的着色进程是以下游方式进行, 应答信息以上游方式传送。所以, 对于初始路径的建立, PD 算法适用于下游自主(DU)模式, CT 算法适用于下游按需(DoD)模式。2) 由于 ATM MPLS 网络不支持 TTL 域, 因而 CT 算法无法利用线程的 TTL 属性来避免着色进程的无限执行, 这是 CT 算法用于 ATM MPLS 网络时的一个局限, 而 PD 算法则不存在这一限制。3) 在 CT 算法中, 中间 ATM-LSR 可对颜色进行合并, 因而比 PD 算法

更能有力地支持流合并机制。4) 因为 PD 算法中传递的路径向量长度不固定, 且受到网络规模的影响, 因而经过的 ATM-LSR 越多, 路径向量长度就越长, 消耗的网络资源也越大; CT 算法在协议中携带的状态信息长度固定, 与网络大小无关, 所以 CT 算法比 PD 算法的扩展性要好。5) 如果通过上路一个结点来关注重构路径的性能, 则无论标记分发模式如何(DU 或 DoD), PD 算法都显示出比 CT 算法更好的性能。这是因为 CT 算法需要额外的信息来维护在每一链路的跳数, 因而 CT 算法需要更多的控制信息。6) 如果通过下路一个结点来考察路径重构的性能, 则在形成一个暂态环路时, CT 算法显示出更好的性能。这是因为一旦出现环路, 大量结点将变为另一结点的上游结点, 而在 PD 扩散计算期间, 请求信息在上游方向上被分发, 这便导致了控制信息数量的增加。

5 结 论

本文经理论分析和基于 ATM 环境下的仿真实验, 得出两种状态归并环路防止级控制机制的比较结果: 对于初始路径的建立, ATM MPLS 网适于采用 CT 算法; CT 算法的扩展性更强; 在 ATM MPLS 网络中, 使用 CT 算法存在无法利用线程的 TTL 属性避免着色进程无限执行的局限; 上路一结点时, PD 算法具有更好的性能; 下路一结点时, CT 算法性能更佳; CT 算法能更好地支持流合并机制。

参考文献(References):

- [1] Thomas M Chen, Tae H Oh. Reliable services in MPLS [J]. *IEEE Communications Magazine*, 1999, 22(12): 53-62.
- [2] Anathan A. Evolution of multiprotocol label switching [J]. *IEEE Computer Communications*, 1998, 20(5): 165-173.
- [3] A Ghanwani, B Jamoussi, D Fedyk. Traffic engineering standards in IP networks using MPLS [J]. *IEEE Communications Magazine*, 1999, 22(11): 49-53.
- [4] Fan Z. New trends in ATM networks: A research view [J]. *Computer Communications*, 1999, 22(2): 499-515.
- [5] Hairong Sun, Ke Huang, Lem in Li. Supporting IP on the ATM networks: An overview [J]. *Computer Communications*, 1998, 21(2): 1020-1029.
- [6] Y Ohba, Y Katsube, E Rosen, et al. MPLS loop prevention mechanism [EB/OL]. <http://www.china-pub.com/>, 2001/2001-06-20.