

基于 NS 的 MPLS 组播的仿真实现* Simulating of MPLS Multicast Based on NS

李睿妍, 李陶深**, 陈 燕

LI Rui-yan, LI Tao-shen, CHEN Yan

(广西大学计算机与电子信息学院, 广西南宁 530004)

(School of Computer, Electronics and Information, Guangxi University, Nanning, Guangxi, 530004, China)

摘要:针对现有的 NS 网络仿真器不能直接对 MPLS 组播进行仿真的情况,通过对 NS 中的 MPLS 模块进行扩展,使 NS 较好地支持对 MPLS 组播的仿真。网络仿真实验表明,本文的方法是有效的而且是可行的。

关键词:组播 网络仿真器(NS) MPLS MPLS 组播

中图分类号:TP393 **文献标识码:**A **文章编号:**1002-7378(2006)04-0284-03

Abstract: To solve the problem that the current NS simulators can not be used directly to simulate multicast protocols in the Multiple Protocol Label Switch(MPLS) network, an approach is presented. In the present method, the MPLS module is extended, and that makes NS to support the simulation of MPLS multicast. The network test shows that this approach is effective and feasible.

Key words: multicast, network simulator(NS), MPLS, MPLS multicast

MPLS(Multi Protocol Label Switching)多协议标签交换技术是继 IP 技术以来的一种在开放的通信网上利用标签引导数据高速、高效传输的新技术,它可以充分利用 ATM 网络的各种资源,实现 IP 分组的快速转发交换。MPLS 在一个无连接的网路中引入连接模式,从而减少了网络复杂性,并且兼容现有各种主流网络技术,能大大降低网络成本,能在提高 IP 业务性能的同时,确保网络通信的服务质量和数据传输的安全性。

组播和 MPLS 是两种互补的技术,在 MPLS 网络上构建组播树可以有效地改进网络的性能,解决 IP 组播的可量测性与过量负载的问题。组播技术与 MPLS VPN 技术的结合将为运营商提供一个前所未有的多业务发展空间。

国外在 MPLS 组播的问题上提出了多种方法。

文献[1]提出一种采用数据驱动的方法,把每个数据源用标签指定到共享树的流量中,以解决在 MPLS 域中支持密集模式和稀疏模式组播的问题;文献[2]提出一种通过控制消息去建立组播树的方法,解决了组播流量问题;文献[3]提出一种对 CR-LDP 的扩展,以便在 L2 层上建立组播树的新方法;文献[4,5]提出为 MPLS 服务扩展 LDP 和 RSVP 的方法等。本文针对现有的网络仿真器(NS)不能直接对 MPLS 组播进行仿真的情况,在 LINUX9.0 环境下利用 NS 实现对 MPLS 模块的扩展,使其较好地支持 MPLS 组播。

1 基于 NS 的 MPLS 组播的仿真设计

在 MPLS 中应用组播技术的核心问题是组播树的构建,即如何将组播树映射为标签交换路径 LSP(Label Switched path)树,并且利用建立的 LSP 转发组播数据包。

NS 是由美国加州 Lawrence Berkeley 国家实验室等单位开发的开源免费网络仿真软件,已广泛应用于局域网、广域网、无线移动网和卫星网络仿真。但是,NS 中的 MPLS 节点不支持组播路由,原因在

收稿日期:2006-07-17

作者简介:李陶深(1957-),男,广西邕宁人,教授,主要从事网络安全、网络路由、分布式数据库方面的研究。

* 本文得到广西留学回国人员科学基金项目(桂科回 0342001)和广西电子信息应用项目(桂电办 2004-17 号)联合资助。

** 通讯联系人。

于:(1)没有针对组播组的标签建立机制;(2)MPLS Classifier 没有组播数据包的复制功能;(3)MPLS 数据头中的指针不能和组播数据包复制器一起工作。为了使 MPLS 节点支持组播,需要对 MPLS 节点做相应的扩展。

1.1 标签分发

通过在各个 LSR 节点的 LDP 代理之间交换 LDP 信息进行标签分发和建立 LSP 路径,可使 MPLS 网络仿真器支持控制驱动、数据驱动、显示路由等三种标签分发模式。

1.1.1 控制驱动模式 该模式在所有 LDP 代理之间分发 LDP 信息,即使没有数据要传输。每个 LDP 代理向其他所有代理发送映射消息(包含 FEC 和标签信息),为每个 FEC 预先建立 LSP 路径,便于以后到达的数据传输。最后,所有 MPLS 节点的 LIB 表被填充,不同的 LSP 路径被分配给所有的 FEC,即使没有数据传输也要对 LIB 表进行填充。

1.1.2 数据驱动模式 该模式仅在要求传送数据的源节点和目的节点之间分发 LDP 消息,并根据 FECs 构建 LSP 路径。当一个节点希望传输数据时,它将把请求消息也发送给上游 FEC。当 FEC 收到请求消息,它将把下游的映射消息发送给源,途中经过的每一个路由将会收到映射消息,对消息处理后把一条新的 LDP 消息发送到沿着向源的方向的下一跳。通过这种方式,就可构造一条从源节点到目的节点的 LSP 路径。

1.1.3 显示路由模式 在该模式中,LSP 路径用一种简单的方式建立。数据包将沿着用户自定义的一系列显示路由节点建立 LSP 路径,映射消息仅仅沿着这个路径被传输来构建相应的 FEC 的 LSP。

1.2 LSP 路径的释放和撤消

上述三种标签分发模式都是建立 LSP 路径的方式。相应地,在 NS 中使用 `ldp-trigger-by-release(fec,LSPid)` 函数可以释放已建立的 LSP 路径。当要释放一个由数据驱动或控制模式驱动建立的 LSP 路径时,这个函数将会把 `LSPid` 设为 `-1`;而当释放一个由显示路由模式建立的 LSP 路径时,函数不仅要重新设置 FEC,还要重新设置 `LSPid`。此函数在某个节点上执行时,这个节点将会根据 FEC 删除信息基表相应的出标签(若是显示路由路径,`LSPid` 也要删除),并发送一个释放消息到相应出接口的下一跳。当下一跳收到释放消息(包含 FEC 和 `LSPid` 信息)后,它将清除它的出标签,并把释放消息发送给它的出接口,直至到达 FEC(目的地址)或非 LSR 节

点。用这种方式,LSP 路径将被释放,标签被回收处于空闲状态以便将来再次分配。

LSP 路径的撤消是另一种标签释放的方式。可在 NS 中使用 `ldp-trigger-by-withdraw(fec,LSPid)` 函数去撤消 LSP 路径。在某个节点上执行此函数时,这个节点将会发送撤消消息到所有与 `(fec,LSPid)` 对应的入接口节点,查询节点信息基表,清除标签。当上游节点收到撤消消息时,它们将会释放它们的出口标签,如此循环,直至到达源节点。使用这种方式,对应 `(fec,LSPid)` 的所有上游标签和 LSP 路径将被删除。

1.3 MPLS 标签交换

当一个数据包到达某节点时,MPLS 分类器将它进行分类、处理和把它转发到一个本地代理或另一节点。数据包的处理按照下面的算法进行处理。

步骤 1:接收数据包,检查数据包头;

步骤 2:如果数据包的包头 `size == 0`,则没有任何标签,为 IP 节点,转步骤 3;否则为标签了的 MPLS 节点,转步骤 4。

步骤 3:假设把目的地址作为 FEC,在 PFT 表里查询 LSP 路径是否已经存在。如果存在,则建立一个 MPLS 头,把出标签和 FEC 插进 MPLS 头,进行 L2 数据包转发;否则判断是否是数据驱动模式,如果是数据驱动模式则必须进行标签分发,否则进行 L3 层路由转发。转到步骤 7。

步骤 4:根据(入标签,入接口)查询 LIB 表,如果存在相应表项,则做标签交换(L2 层转发),转步骤 5;否则做 L3 层路由,转步骤 6。

步骤 5:如果出标签 `= 0`,把 MPLS 头弹栈;如果出标签 `> 0`,交换 MPLS 头;如果出接口不存在,回到 L3 层路由。转到步骤 7。

步骤 6:基于使用的路由协议,使用 IP 源和目的应用 L3 层路由。

步骤 7:算法结束。

1.4 组播树的复制和转发

MPLS 组播数据包的转发在非分枝节点和单播数据包的转发完全一样,在分枝节点,转发算法要为每一个出接口复制接收到的数据包,而在分枝节点 LIB 包含多个输出表项,对应与每一个出接口都需要复制一份数据包,非分枝节点转发过程只需要处理一个出接口。为了实现数据的复制与转发,可在 MPLS 模块 `classifier-addr-mpls.cc` 中添加如下函数:

```
int MPLSAddressClassifier::LIBgetAll(int
```

entrynb,

```
int &iiface,int &iLabel,int &oiface,
int &oLabel,int &LIBptr)
```

在 LIB 表中能以 iLabel 检索到 n 个(oiface,oLabel), 以得到多个出接口,进行数据包复制。

1.5 TCL 映射

通过添加新的基本网络对象可以扩展 NS,但是既然由于效率的原因,数据通道上的对象类都是用 C++实现的,那么通常就需要在 C++代码中做 OTCL 映射。具体的做法是在 classifier-addr-mpls. cc 的命令函数中添加与相应的 OTCL 对象中的成员函数一样的命令。

2 MPLS 组播的仿真测试

在 LINUX 系统下安装的 ns-allinone-2.29 版本的网络仿真器下进行 MPLS 组播仿真测试。按照前面所述的设计修改 NS 里的源程序,对 MPLS 模块进行组播扩展,编译通过后就可以模拟指定信源的 MPLS 组播。仿真实验的拓扑结构如图 1 所示。假设信源在节点 0 并且设置了组地址,节点 1、节点 2、节点 3 为组成员,节点 4 为分枝节点。在信源开始传输数据之前,先让节点 1 和节点 2 在 0.5s 加入组 (S, G), 拓扑结构如图 2 所示。在 1.0s 数据源开始发送数据包,在 1.2s 节点 1 离开组播组,1.3s 节点 3 加入组。

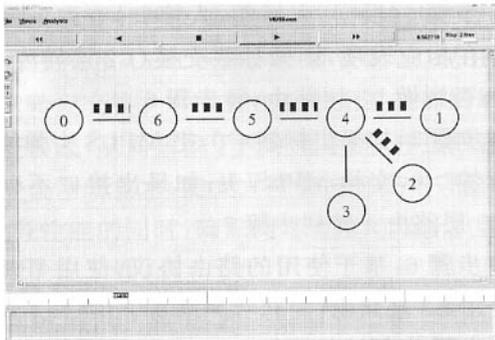


图 1 MPLS 组播实验的拓扑

图 3 是仿真测试的结果,其中 Y 轴表示数据包到达的时间,X 轴表示到达节点 3 的包序列。图 3 结果说明,利用 MPLS 技术的快速转发功能可以明显减少从信源 0 到达节点 3 的组播包的传输时延。

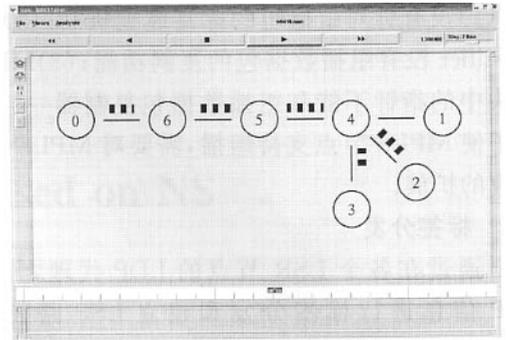


图 2 结点 1 退出、节点 3 加入后的 MPLS 组播拓扑

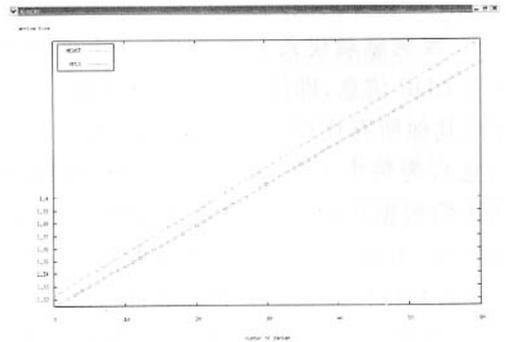


图 3 模拟仿真结果

3 结束语

多协议标记交换技术(MPLS)是一种在开放的通信网上利用标签引导数据高速、高效传输的新技术。本文针对现有的 NS 网络仿真器不能直接对 MPLS 组播进行仿真的情况,通过对 NS 中的 MPLS 模块进行扩展,使 NS 较好地支持对 MPLS 组播的仿真。网络仿真实验表明,本文的方法是有效的而且是可行的。

参考文献:

- [1] ACHARYA A, GRIFFOUL F, ANSARI F. Ip multicast support in mpls networks[R]. IETF internet draft, February 1999.
- [2] LEE C Y, ANDERSSON L, CARLBERG K, AKYOL B. Engineering paths for multicast traffic [R]. IETN Internet draft, October 1999.
- [3] OOMS D, HOEBEKE R, CHEVAL P. Mpls multicast traffic engineering. IETF Internet draft [R]. February 2001.
- [4] CHUNG J, BENITO M, CHHABRA H, CHO G, RASIAH P. Ldp extensions for mpls multicasting services[R]. IETF Internet draft, February 2002.
- [5] CHUNG J, BENITO M, CHHABRA H, CHO G, RASIAH P. Rsvp-te extensions for mpls multicasting services[R]. IETF Internet draft, February 2002.

(责任编辑:邓大玉)