文章编号: 1001- 2486(2008) 02- 0051- 05

# 一种提高应用层组播转发速率的机制*

曹继军, 苏金树, 吕高锋
(国防科技大学 计算机学院, 湖南 长沙 410073)

摘 要: 针对应用层组播报文转发的特点, 提出了一种能够提高应用层组播转发速率的新机制。该机制降低了应用层组播报文从主机内存到网卡缓冲区之间数据复制的次数, 节省了 CPU 处理开销。理论分析表明该机制能够降低应用层组播延迟和提高应用层组播转发速率。实验验证了该机制的可行性与有效性。

关键词: 应用层组播; 转发速率; 网卡; 延迟; 代理服务器
中图分类号: TP393 文献标识码: A

# A Mechanism Improving the Forwarding Throughput of Application Layer Multicast

CAO Ji-jun, SU Jin-shu, LU Gao-feng
(College of Computer, National Univ. of Defense Technology, Changsha 410073, China)

**Abstract**: Based on the characteristics of packet forwarding in application layer multicast (ALM), this paper proposes a high performance multicast mechanism, in which a NIC-based multicast mechanism is used to send multiple replicas of an ALM packet to different destinations with less CPU intermediation in comparison to the traditional host-based multicast mechanism. By adopting the mechanism, the CPU overhead and the number of times data packets are copied from host memory to NIC buffer. Theoretical analysis indicates that the new mechanism can reduce the delay and improve the forwarding throughput in application multicast. We have modified the NIC driver and added some corresponding APIs to network protocol stack in order to implement the proposed mechanism. Experimental results also indicate the feasibility and validity of this mechanism.

**Key words**: application layer multicast; forwarding throughput; NIC; delay; proxy

组播是一种重要的数据传送方式, 最早提出的组播实现方案是 IP 组播[1]。IP 组播具有最高的传输效率[2], 但是由于技术和市场等原因, 迄今为止 IP 组播并没有得到广泛部署[3]。为此研究者提出应用层组播(ALM: Application Layer Multicast)技术。应用层组播的核心思想是将对组播的支持从网络核心转移到终端系统, 这符合"End-to-End Argument[4]"所提倡的互联网设计原则。相对于 IP 组播, 应用层组播最大的优势在于易于部署。目前的应用层组播研究提出两种主要体系结构: 对等型结构和代理型结构。前者完全由端系统构成, 所有组播功能都是在参与组播会话的端系统中实现, 后者由应用层组播服务器和端系统组成[5]。

研究者提出了多种应用层组播协议[5-8], 各种协议都以构造应用层组播转发树为目标, 很少针对应用层组播转发机制开展研究。通过深入分析应用层组播报文转发的特点, 本文提出一种新的支持应用层组播转发的机制, 该机制能够显著提高应用层组播转发速率。

## 1 基于网卡的应用层组播

### 1.1 应用层组播报文转发特点分析

应用层组播传输数据的过程为: 组播源节点产生组播数据报文, 查询应用层组播路由表以获得其子节点的单播地址列表, 再分别以这些单播地址为目的地址发送报文。报文通过单播路径传输给子节点,

* 收稿日期: 2007- 09- 10
基金项目: 国家自然科学基金重点资助项目(90604006); 国家部委资助项目
作者简介: 曹继军(1979—), 男, 博士生。

子节点接收报文后查询应用层组播路由表,再分别向自己的各个子节点转发报文。接收到组播报文的节点重复此过程,直到所有节点接收到组播报文为止。

通常,应用层组播节点处理数据报文的过程为:(1)当报文到达网卡时,网卡将接收到的报文保存在网卡缓冲区,然后向主机 CPU 通告主机接收中断,主机 CPU 设置包含主机内存地址和需要传输的字节数的 DMA 寄存器,并启动从网卡缓冲区到主机内存的数据传输过程。一旦数据传输完成,DMA 控制器将向 CPU 通告数据传输完成中断。CPU 接收到 DMA 的中断后开始处理数据;(2)当主机通过网卡发送数据报文时,主机 CPU 设置包含主机内存地址和需要传输的字节数的 DMA 寄存器,并启动从主机内存到网卡缓冲区的数据传输。一旦数据传输完成,DMA 控制器将向网卡控制器通告数据发送中断。接着网卡开始对数据报文进行处理并且最终将数据报文发送到网络。主机应用程序设置主机内存中数据报文的目的地址信息,然后将数据报文从主机内存复制到网卡缓冲区,网卡将报文封装成特定格式的数据帧并且将其发送到网络。假设该节点有 $n$ 个子节点,则上述过程要重复完成 $n$ 次,即要对数据内容相同、仅仅是目的地址不同的数据报文进行 $n$ 次从主机内存到网卡缓冲区的复制。

### 1.2 支持应用层组播的新机制

基于以上分析,本文提出了一种支持应用层组播的新机制。新机制的基本思想为:主机将组播数据报文的内容和该节点的子节点地址信息发送给网卡,由网卡负责为数据报文设置目的地址信息并且将其发送到网络。本文称采用了新型组播转发机制的应用层组播为基于网卡的应用层组播,而将传统的应用层组播称为基于主机的应用层组播。

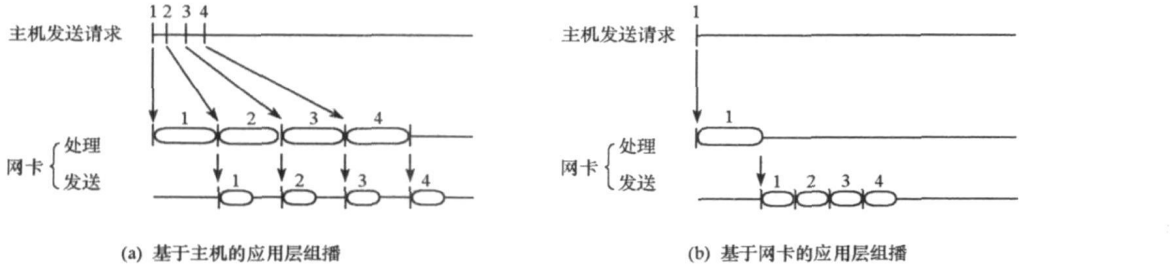


图 1 基于主机和基于网卡的应用层组播处理时序图

Fig. 1 The timing diagrams comparing host-based ALM and NIC-based ALM

图 1(a)为基于主机的应用层组播时端系统向 4 个子节点发送报文的时序图。图中所示的三条平行直线代表三个步骤:(1)主机 CPU 产生 4 个报文发送请求;(2)网卡 DMA 顺序处理请求,即数据报文从主机内存复制到网卡缓冲区并且进入发送队列等待发送;(3)网卡完成数据报文的发送。可见,网卡需要重复 4 次步骤(2)的操作。如果网卡发送报文的过程不能隐藏处理请求过程的延迟,这将导致很高的处理延迟。图 1(b)为基于网卡的应用层组播时端系统向 4 个子节点发送报文的时序图。主机只需要产生一次发送请求,网卡维护组播地址列表并且向列表中的所有子节点发送组播报文。

## 2 新机制的优点分析

可见,对于应用层组播端系统而言,基于网卡的应用层组播机制能够减轻单个主机 CPU 的负载和降低组播数据报文的发送延迟,从而提高端系统的组播转发性能。而应用层组播是由多个端系统参与的群组通信,下面将分析这种新机制对整个应用层组播的益处。

### 2.1 应用层组播时延模型

组播延迟成为衡量应用层组播质量的重要标准。源节点的发送时间、中间节点的接收和转发处理时间以及叶节点的接收时间统称为节点处理延迟。应用层组播父子节点间的数据传输是基于 IP 单播的,期间可能经历多台网络设备以及多跳物理链路,网络设备的转发延迟和物理链路传输延迟统称为节点间的传输延迟。

我们将单源应用层组播树映射成有向图 $G(V, E)$,其中集合 $V$ 中的顶点代表参与应用层组播的端

系统;集合 $E$ 中的边代表应用层组播节点间的单播路径,表 1 列出了相关参数。

表 1 应用层组播时延模型相关参数说明

Tab. 1 Related parameters of ALM delay model

| 参数 | 描述 |
|---|---|
| $D$ | 应用层组播延迟,也称为"树"延迟,即应用层组播节点的最大延迟。 |
| $D$ | 应用层组播平均延迟,即接收应用层组播数据报文的所有节点延迟的算术平均值。 |
| $r$ | 应用层组播树的根节点,即应用层组播数据源节点。 |
| $N$ | 应用层组播树的叶节点集合。 |
| $c(u,v)$ | 应用层组播节点 $u$ 和 $v$ 之间的传输延迟。 |
| $p(v)$ | 应用层组播节点 $v$ 的节点处理延迟。 |
| $p_{u,v}$ | 应用层组播节点 $u$ 和 $v$ 之间的路径,常用$< v_1, v_2, \cdots, v_k >$ 表示,其中 $k$ 为该路径上的节点数目。 |
| $l(p_{u,v})$ | 路径 $p_{u,v}$ 的长度,即该路径上单播路径的数目,也就是该路径上的节点数目减 1。 |
| $d(u,v)$ | 路径 $p_{u,v}$ 的延迟,也就是节点 $u$ 到 $v$ 之间的延迟。 |

根据以上的分析,有以下结论成立:

(1) $d(u,v)=\sum_{i=1}^{k-1}[p(v_i)+c(v_i,v_{i+1})]$,其中,$k=l(p_{r,v})+1$,$v_i$ 表示路径 $p_{u,v}$ 上的第 $i(1\leqslant i\leqslant k)$ 个节点,即节点 $u$ 到 $v$ 之间的延迟等于路径 $p_{u,v}$ 上节点之间的传输延迟与节点处理延迟之和;

(2) $D=\max_{v\in N} d(r,v)$,即应用层组播延迟为组播树的根节点到各个叶节点之间延迟的最大值;

(3) $D=\sum_{v\in V(G)-\{r\}} d(r,v)/(|V|-1)$,即应用层组播平均延迟是组播树根节点到其他所有节点的延迟的算术平均值。

## 2.2 对组播延迟的影响分析

为了分析新转发机制对整个应用层组播系统延迟的优化,首先假设存在两种应用层组播的方案(方案 1 和方案 2),而且它们满足如下条件:(1) 方案 1 的所有节点的网卡都是传统网卡,采用基于主机的应用层组播;方案 2 的所有节点的网卡都是使用了新机制的网卡,采用基于网卡的应用层组播;(2) 两种方案的主机 CPU 和内存性能相同,即主机 CPU 发送请求和从主机内存复制报文到网卡缓冲区的时间相同。为了便于理论分析,我们只比较组播树相同时这两种方案的组播延迟。下面讨论中还需要引入表 2 所列参数。

表 2 应用层组播延迟分析相关参数说明

Tab. 2 Parameters for the analyses of ALM delay

| 参数名称 | 含义 |
|---|---|
| $D$ | 基于主机的应用层组播延迟。 |
| $D'$ | 基于网卡的应用层组播延迟。 |
| $\Delta D$ | 应用层组播延迟优化,即 $\Delta D=D-D'$。 |
| $D$ | 基于主机的应用层组播平均延迟。 |
| $D'$ | 基于网卡的应用层组播平均延迟。 |
| $\Delta D$ | 应用层组播平均延迟优化,即 $\Delta D=D-D'$。 |
| $TP(v)$ | 基于主机的应用层组播过程中节点 $v$ 的处理延迟。 |
| $TP'(v)$ | 基于网卡的应用层组播过程中节点 $v$ 的处理延迟。 |
| $\Delta TP(v)$ | 节点 $v$ 的处理延迟优化,即 $\Delta TP(v)=TP(v)-TP'(v)$。 |

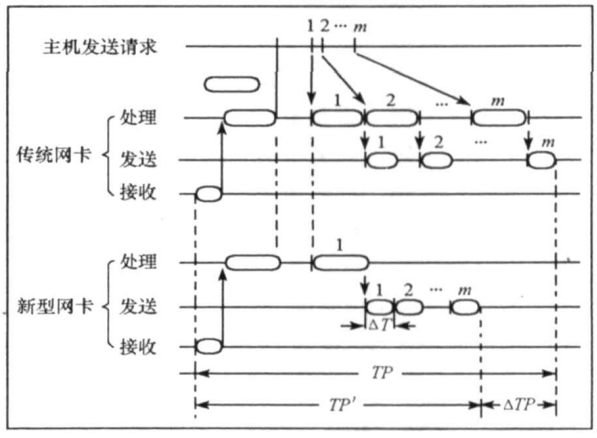


图 2 应用层组播中的处理延迟

Fig. 2 The processing delay of ALM node

表 2 中的部分参数具体含义如图 2 所示。

基于主机的应用层组播延迟是从根节点 $r$ 到叶节点 $v(v\in N)$ 的各个组播路径延迟的最大值,即为

$$D = \max_{v \in N} d(r, v) = \max_{v_k \in N} \sum_{i=1}^{k-1} [TP(v_i) + c(v_i, v_{i+1})] \quad (1)$$

其中, $k = l(p_{r,v}) + 1$, $v_i$ 表示路径 $p_{u,v}$ 上的第 $i$ ($1 \leqslant i \leqslant k$) 个节点。同理, 基于网卡的应用层组播延迟为 (等式中 $k$ 和 $v_i$ 的含义同上)

$$D' = \max_{v \in N} d(r, v) = \max_{v_k \in N} \sum_{i=1}^{k-1} [TP'(v_i) + c(v_i, v_{i+1})] \quad (2)$$

值得注意的是, 上述两种方案的最大延迟路径(即具有最大路径延迟的路径)不一定相同。如果两种方案的最大延迟路径相同(通常是相同的), 那么延迟优化为

$$\Delta D = D - D' = \sum_{i=1}^{k-1} [TP(v_i) - TP'(v)] = \sum_{i=1}^{k-1} \Delta TP(v_i) \quad (3)$$

其中, $k$ 为最大延迟路径上的节点数。所以, 延迟优化为最大延迟路径上各个节点(除叶节点)的处理延迟优化之和。为方便起见, 假设组播树有 $n$ 个性能相同的节点, 而且除叶节点外, 各个节点度均为 $m$; 再假设各个叶节点的深度都相同, 那么在工作负载相同的情况下, 各个节点的处理延迟优化也相同, 设都为 $\Delta TP$, 则此时的延迟优化为

$$\Delta D = \sum_{i=1}^{k-1} \Delta TP(v_i) = (\log_m n) \cdot \Delta TP \quad (4)$$

式(3)和(4)表明, 应用层组播的规模越大, 新机制对应用层组播延迟优化越明显。计算基于主机的应用层组播平均延迟的方法为

$$D = \sum_{v \in V(G) - \{r\}} d(r, v) / (|V| - 1) = \sum_{v \in V(G) - \{r\}} \sum_{i=1}^{k-1} [TP(v_i) + c(v_i, v_{i+1})] / (|V| - 1) \quad (5)$$

同理, 基于网卡的应用层组播平均延迟为

$$D' = \sum_{v \in V(G) - \{r\}} \sum_{i=1}^{k-1} [TP'(v_i) + c(v_i, v_{i+1})] / (|V| - 1) \quad (6)$$

所以, 平均延迟优化为

$$\Delta D = D - D' = \sum_{v \in V(G) - \{r\} - N} \Delta TP(v) / (|V| - 1) \quad (7)$$

由式(7)可知, 平均延迟优化约等于各个节点的处理延迟优化的算术平均值, 它小于单个节点的处理延迟优化。

## 3 实验验证

为了验证本文所提出的新机制的可行性和有效性, 我们在真实的环境中对基于主机的应用层组播和基于网卡的应用层组播进行了对比测试。被测主机的软件配置为 Linux Magic 操作系统, 其内核版本为 2.6.11.12; 硬件配置为 CPU: Intel Pentium ⅣCPU 2.66GHz, 内存: 1GB; 发送报文的网卡为 Intel ⅣPRO/1000 Network Driver, 网卡驱动版本为 e1000"5.6.10.1-k2" DRIVERNAPI。通过配置不同的测试报文长度和子节点数目, 本文测试了在多种情况下基于主机和基于网卡的应用层组播转发报文的最大速率(在不丢失报文前提下)。

测试结果如图 3(a) ~ (d) 分别为测试报文长度等于 64B、128B、256B、和 512B 情况下的测试结果。在每个子图中, 横坐标代表组播转发测试时配置的子节点数目, 纵坐标记录了各个子节点能够获得的最大速率之和。

可见, 当测试报文长度一定时, 基于主机的应用层组播最大转发速率基本恒定, 所以单个子节点得到的最大转发速率与父节点的度成反比; 基于网卡的应用层组播最大转发速率随父节点度的增加而线性增加, 所以单个子节点得到的最大转发速率随父节点度的增加而减小。新机制使得各种测试报文长度下应用层组播转发速率平均提高 40.49% ~ 132.45%。同时可以看出, 总的最大转发速率随着报文长度的增加而增加, 单个子节点得到的最大转发速率也随着报文长度的增加而增加。

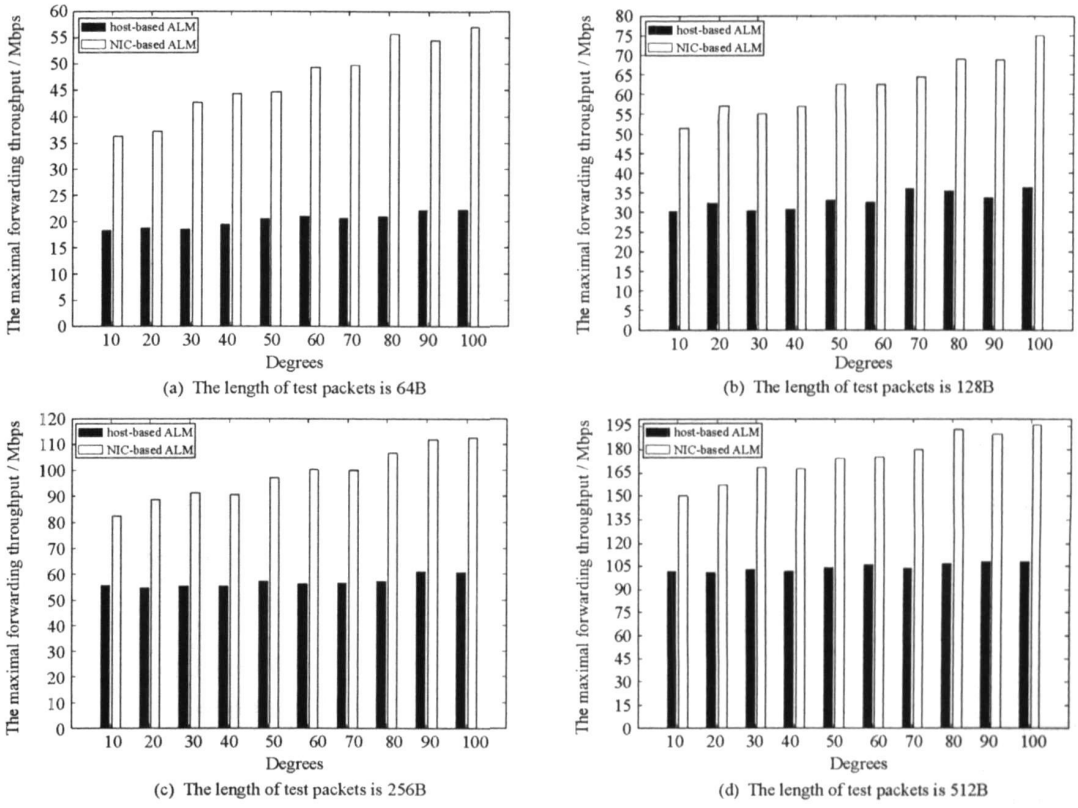


图 3 基于网卡和基于主机的应用层组播转发速率比较

Fig. 3 The comparison of forwarding throughput between NIC-based ALM and host-based ALM

## 4 总结和进一步的工作

随着应用层组播技术的不断发展, 参与应用层组播的端系统数目将会不断增加。基于网卡对应用层组播实施加速, 其实现的代价低, 收益大。支持应用层组播的网卡最直接的优点是减轻了主机 CPU 负载, 降低了节点转发组播报文的处理时间, 这对于减小应用层组播延迟和提高应用层组播转发速率都有很大益处。本文提出的新机制及其实现方法都是针对基于 UDP 的应用层组播的, 如何将该思想扩展到基于 TCP 的应用层组播有待进一步研究。

## 参 考 文 献:

[1] Deering S. Multicast Routing in a Datagram Internetwork[D]. Ph. D. Thesis, Standford University, 1991.

[2] Diot C, Levine B, Lyles J, et al. Deployment Issues for the IP Multicast Service and Architecture[J]. IEEE Network, 2000, 14(1): 78- 88.

[3] Almeroth K C. The Evolution of Multicast: From the Mbone to Inter-domain Multicast to Internet2 Deployment[J]. IEEE Network, 2000, 14: 10- 20.

[4] Saltzer J, Reed D, Clark D. End-to-end Arguments in System Design[J]. ACM Transactions on Computer System(TOCS), 1984, 2(4): 195- 206.

[5] Pendarakis D, Shi S, Verma D, et al. ALMI: An Application Level Multicast Infrastructure[C]// Proceedings of 3rd USEWNIX Symposium on Internet Technologies and Systems(USITS), 2001: 49- 60.

[6] Chu Y, Rao S, Zhang H. A Case for End System Multicast[C]// ACM SIGMETRICS, 2000: 1- 12.

[7] Abhishek S, Azer B, Ibrahim M. dPAM: A Distributed Perfetching Protocol for Scalable Asynchronous Multicast in P2P Systems[C]// Proc. of INFOCOM' 05, 2005.

[8] Radha V, Gulati P. Appcast: A Low Stress and High Stretch Overlay Protocol[J]. Int. J. Grid and Utility Computing, 2005, 1(1): 38- 45.